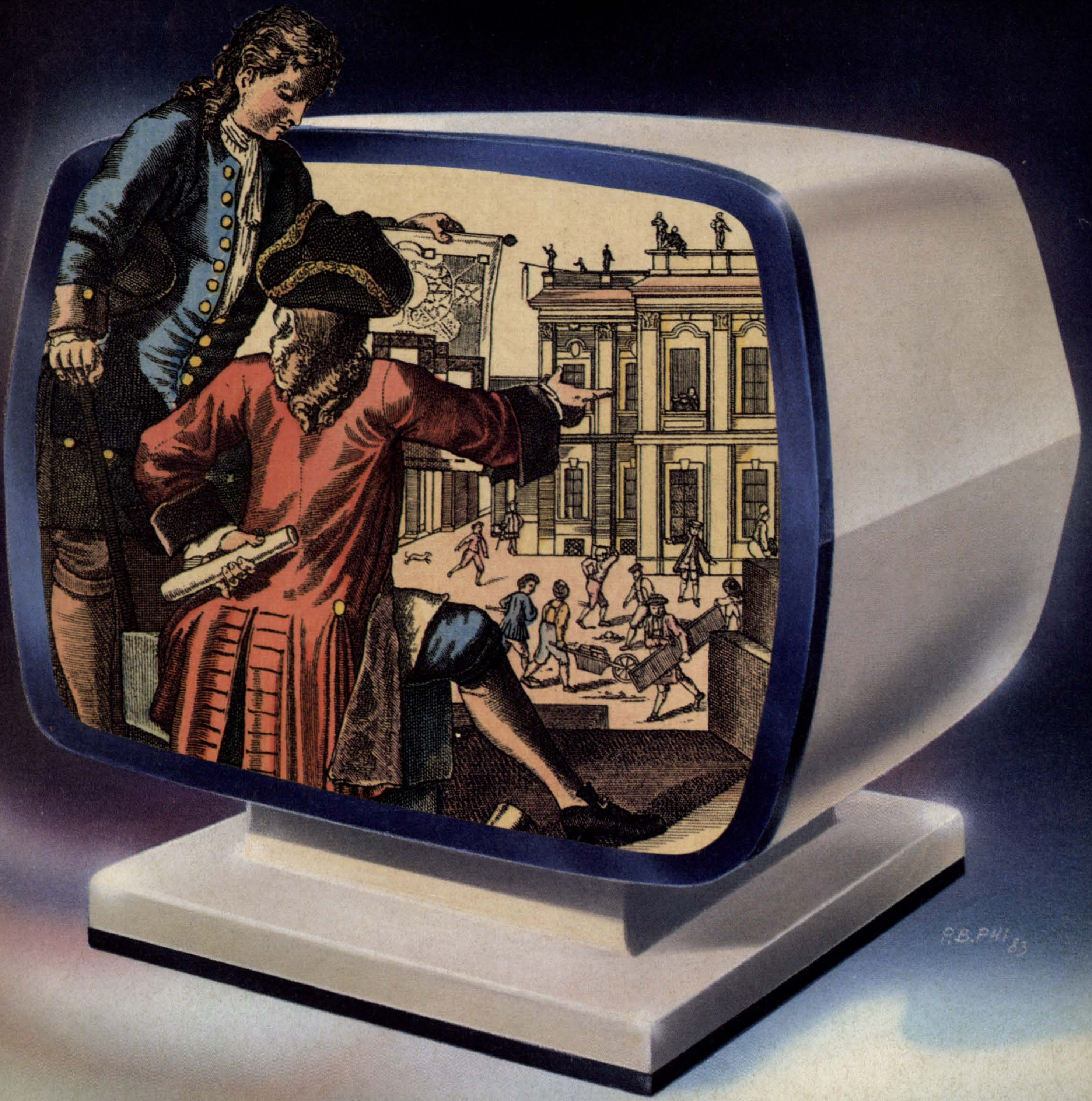
User's Guide™

TO SYSTEMS & SOFTWARE

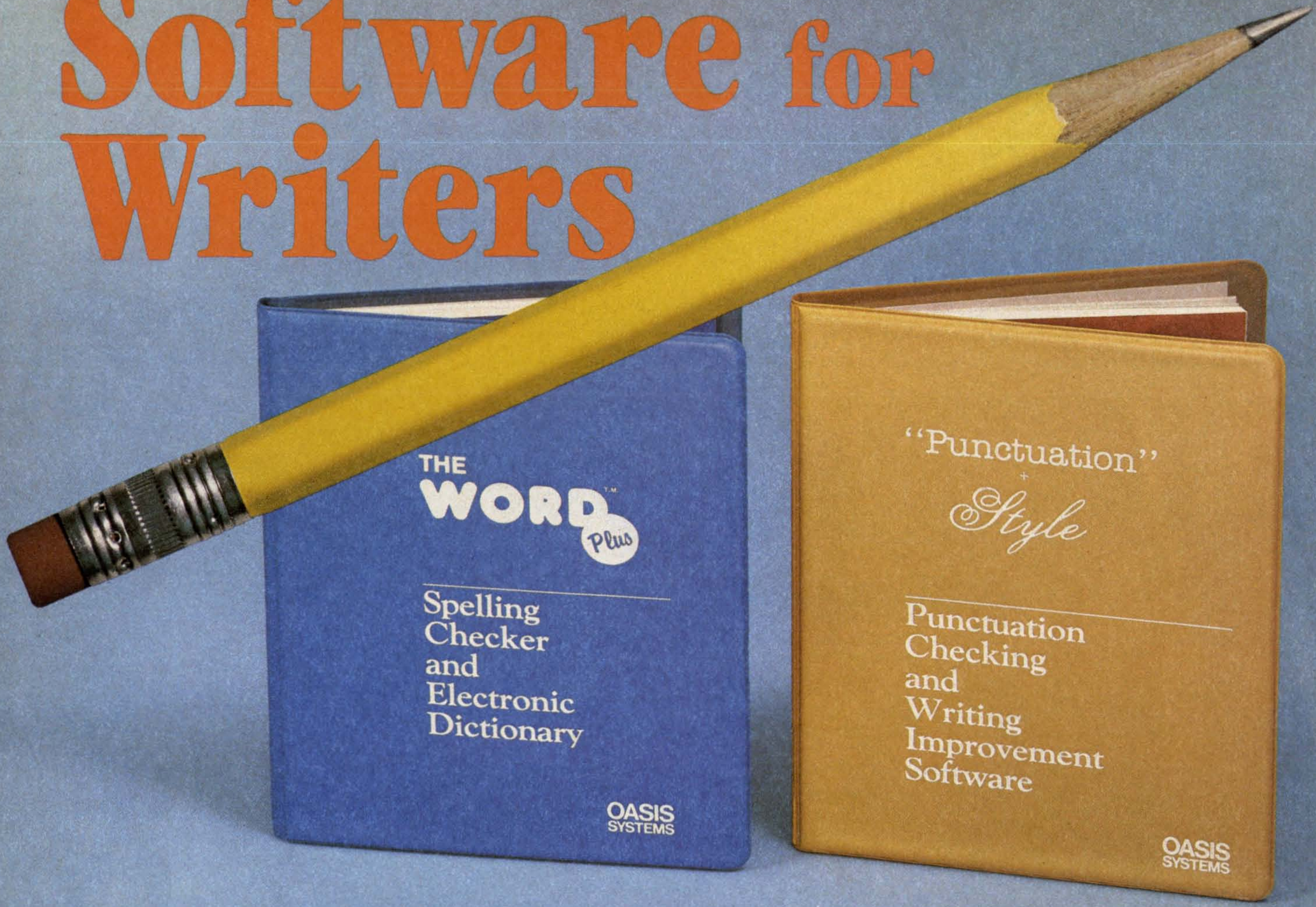
NO. 5 $4

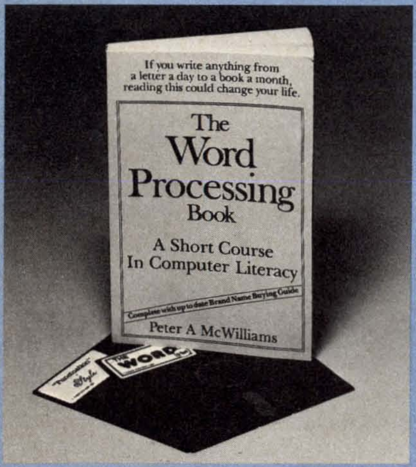
CP/M Plus Tutorial dBASE II Tutorial Perfect Filer

Radio Shack Model 100 PMC MicroMate TeleVideo & WordStar Hard Disks



Software for Writers


THE WORD™ Plus
Spelling Checker and Electronic Dictionary
OASIS SYSTEMS
"Punctuation" + Style
Punctuation Checking and Writing Improvement Software
OASIS SYSTEMS


If you write anything from a letter a day to a book a month, reading this could change your life.
The Word Processing Book
A Short Course In Computer Literacy
Complete with up-to-date Brand Name Buying Guide
Peter A McWilliams
WORD

"Oasis Systems' software — Unquestionably the best". . .

Peter McWilliams, author of the #1 best-selling book on word processing.

The WORD Plus is the standard by which other spelling checkers are measured. Here's why:

- *Real* 45,000 word dictionary.
- Shows errors "in-context."
- Interactive word look-up finds correct spelling for you and corrects at the push of a button.
- Hyphenates words automatically.
- Solves crosswords, puzzles, and anagrams.
- Works with almost any CP/M®, CP/M-86® or MS/DOS (1.0, 1.1 and 2.0) compatible word processing program (WordStar, Magic Wand (PeachText), Spellbinder, Perfect Writer, Select, Final Word, Volkswriter, . . . and more!).

Punctuation & Style takes the worry out of writing by automatically catching dozens of different punctuation errors, both common and obscure. In addition, P&S catches unpaired format commands (underline, boldface, etc.), doubled words, and more.

P&S gives you a "critique" of your writing, suggesting alternatives for commonly misused or over-worked phrases. It also shows where active voice can replace passive voice to add clarity and precision.

Punctuation & Style is the perfect companion to The WORD Plus. It works easily with most CP/M word processors. (Available soon for CP/M-86 and MS/DOS.)

Call or write for complete information: 619-222-1153

OASIS SYSTEMS

2765 Reynard Way
San Diego, CA 92103

Circle number 20 on the reader service card.

Dealers contact:
SOFTWARE DISTRIBUTORS
1-800-252-4024 (in California)
1-800-421-0814 (outside California)

CP/M and CP/M-86 are registered trademarks of Digital Research, Inc.

User's Guide
TO **CP/M**® SYSTEMS & SOFTWARE

User's Guide **Volume 1, Number 5**
Sept. 1983.

User's Guide , ISSN 0745-4791, is published six times a year by TUG, Inc., Box 3050, Stanford, CA 94305. Second-class postage paid at Redwood City, CA 94063 (USPS 689-550).

Entire contents Copyright © 1983 by TUG, Inc. All rights reserved. Reproduction of material appearing in *User's Guide* is forbidden without explicit permission. Send all requests to The Editors, Box 3050, Stanford, CA 94305.

SUBSCRIPTION INFORMATION:
Single copy price $4.00 ($5.50 sent first-class). Subscription for one year is: U.S. — $18 if payment accompanies order, $21 if subscription is billed. Add $10 for first-class delivery.
FOREIGN — Single copy price $5.50. Subscription for one year (surface mail) is $24 (U.S. currency) if payment accompanies order, $29 (U.S. currency) if subscription is billed. Inquire about rates for airmail delivery outside the U.S.
Inquiries about bulk, retail sale, and group rates are encouraged. Address subscription inquiries to *User's Box 3050, Stanford, CA 94305. Allow 6-8 weeks for processing of order.*

POSTMASTER: Please send address changes (Form 3579) to User's Guide, Box 3050, Stanford CA 94305.

About the Cover

This issue's inspiration, an anonymous German watercolor of the 17th Century titled "Der Architect," was computerized by Phi Bach Pham.

CP/M and CP/M Plus are trademarks of Digital Research.
Cache/Q is a trademark of Techne Software Corp. (Lafayette, CA).
WordStar is a trademark of MicroPro Int. (San Rafael, CA).
dBASE II is a trademark of Ashton-Tate.
CP+ is a trademark of Taurus Software (Lafayette, CA).
The Organizer is a trademark of The Information People (Newark, OH).
MicroMate is a trademark of Personal Micro Computers, Inc.(Mt. View, CA).
Radio Shack is a registered trademark of Tandy Corp. (Fort Worth, TX).
User's Guide and Impatient User's Guide are trademarks of TUG, Inc.

Contents Vol.1, No.5 Sept 1983

Letters... 7
Editorial... **Rumors** 2

For Starters... **Laying a Foundation** 6
Setting up disk storage for data base applications.

Tutorial... **An Introduction to dBASE II** 8
by Bob Cowart
The most popular data base management program for CP/M systems and IBM Personal Computers.

Tutorial... **Using CP/M Plus, Pt. 1: Commands and File Utilities** 16
by Tony Bove & Cheryl Rhodes
Part one: from system start-up and command line editing to file attributes, passwords and utilities.

Application Notes... **Subscription Fulfillment With dBASE II : Part 1** 32
by Tony Bove & Cheryl Rhodes
An overview of data entry, editing, retrieval, listings and mailing label printing.

Overview & Starter... **Perfect Filer** 44
by Matthew McClure
A data base program bundled with Kaypro systems and available for CP/M and IBM PC-compatible systems.

For Experienced Users... **CP/M Plus: Inner Workings** 50
by Steve Rosenthal
How CP/M Plus controls extra memory.

Keep Ahead of the Jargon Dept.... **Hard Disks For Small Computers** 56
by Steve Rosenthal
Hard disk technology unveiled and explained.

Overview... **TeleVideo 950 & WordStar EPROM** 62
by Tony Bove & Cheryl Rhodes
WordStar commands using the function keys of a TeleVideo 950 terminal.

System Overview... **The Radio Shack Model 100 Portable** 66
by Dan Dugan
A "lap" computer suitable for writing, note-taking and daily messages.

System Overview... **The PMC MicroMate With CP/M Plus** 72
by Tony Bove & Cheryl Rhodes
A transportable system that works with any terminal.

STAT... **The Answer Man** 78
by Kelly Smith
Answers to recent questions, and advice on "flippies."

Product Announcements... Systems, software, peripherals 77
Advertiser Index... 43

EDITORIAL Tony Bove & Cheryl Rhodes (editors),
Kelly Smith, Steve Rosenthal, Dan Dugan (contributing editors)
ART Phi Bach Pham (cover), Contributors:
Lon Shoemaker (cartoons), Rose Skytta (photos)
PRODUCTION Carol Hartung, Anita Rhodes, Lon Shoemaker
CIRCULATION William Shoemaker, Cathy Slusser
ADMINISTRATION Carol Hartung
Special Thanks: Dave Parkinson (Technoweb),
Bob Cowart, Matthew Mcclure, Eugene Wulsin.
Typesetting by Bove & Rhodes with George Graphics, San Francisco, and Abra Type, Palo Alto. Camera work by Camera/Graphic, Palo Alto. Printed by TechnoWeb, San Jose.

Rumors

IBM, Texas and the Whole Earth

Rumor has it that IBM will introduce its own operating system. Rumor has it that Texas Instruments plans to introduce an under-$1000 computer featuring Concurrent CP/M in ROM. Another rumor has Digital Research, Inc. (makers of CP/M family of systems) and VisiCorp (makers of VisiCalc and Visi-series) negotiating a merger, to become what one industry expert called "the largest software company in the world."

Ahh, rumors. This industry has more "rumor-mongers" and "moles" than Georgetown, and *InfoWorld* covers the scene like a computer-wise *Washington Post*. The information may be exciting to heavy industry-watchers, but out of context it is meaningless to the majority of computer users.

What does it mean to you that DRI and VisiCorp might merge? It means that VisiOn, announced earlier this year to be available this October, should appear on sixteen-bit CP/M-86 systems. What does a new IBM operating system mean? Another major compatibility issue for IBM PC users. What does an under-$1000 CP/M-in-rom computer mean? CP/M systems without a need for floppy disks; perhaps CP/M will also appear on "lap" computers.

These are guesses at best. More often than not, a major technological decision is not felt in the marketplace for at least a year, until it manifests itself in the most visible computers on the market. However, rumors abound in the press and sometimes they force companies into "pre-announcing" non-existing products.

We can provide the unvarnished truth about one rumor making the rounds this summer: that *User's Guide* is expanding its content to include the IBM Personal Computer.

Editorial Plans

You may have noticed the subtle change in our cover logo. We've "now included" the IBM PC (and PCDOS) in our coverage of systems and software. Fact is, we've always included the PC.

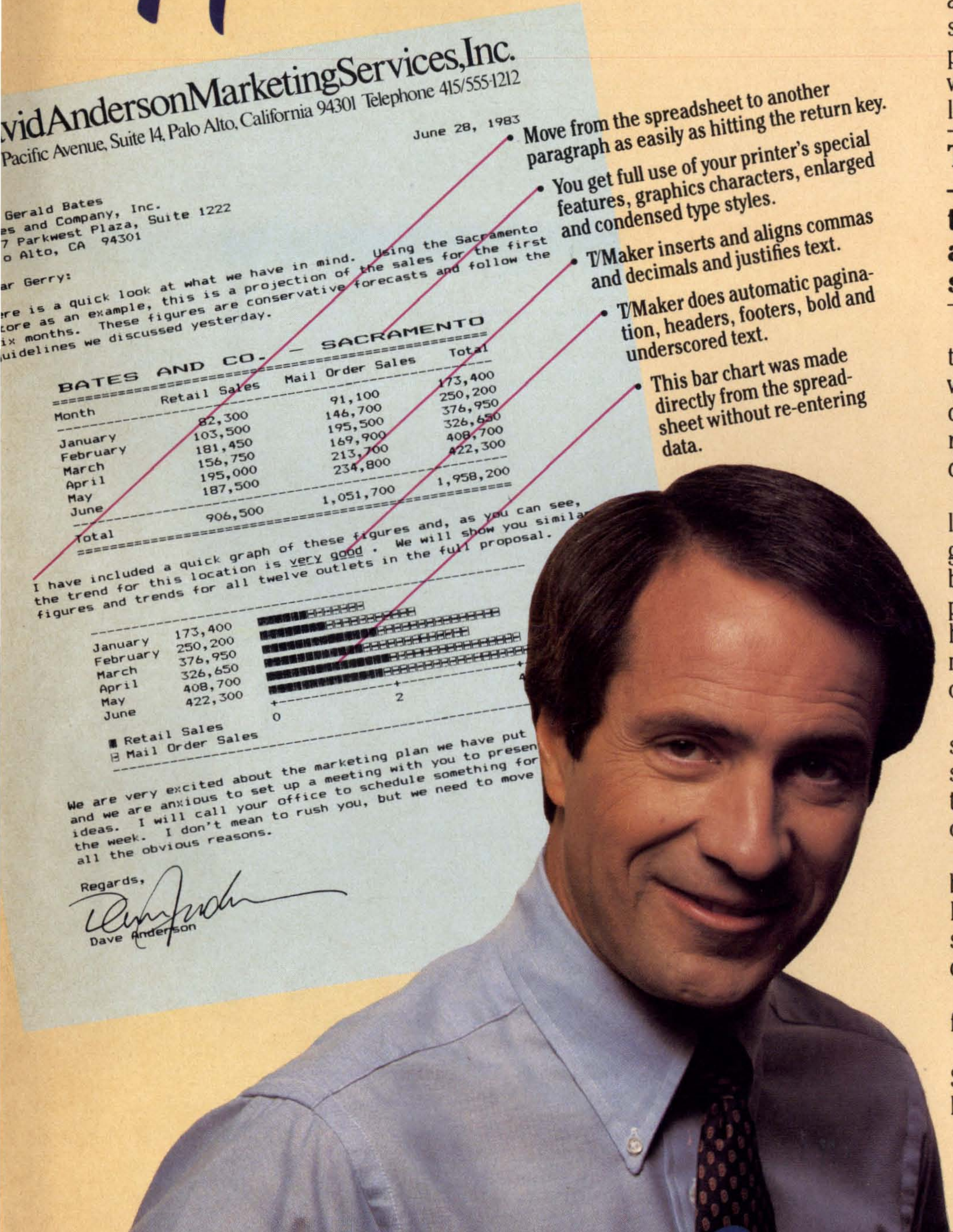
User's Guide has not dropped coverage of CP/M; on the contrary, we've beefed up our coverage of the system itself, as well as the application programs that run on it. The fact is that many people think that we don't cover the IBM PC. Given the fact that we included an overview of the PC in our first issue, we couldn't understand why there was such a misconception.

Parlor talk convinced us that many users think they are either on the CP/M side of the world or the PCDOS side (not to mention Apple SOS, Radio Shack TRSDOS, UNIX, Oasis and other sides). Random conversations among computer users showed that many thought CP/M was only for eight bit systems, or that PCDOS application software would prevail over CP/M application software.

This seemed absurd, with WordStar and dBASE II (two popular CP/M-based programs) topping the list of the most popular IBM PC programs, running under PCDOS. Much of the application software for the IBM PC evolved from CP/M versions.

User's Guide has been running articles about WordStar, dBASE II, CP/M-86, SuperCalc and other programs, some of which are on the list of the most used programs on the IBM

T/Maker



vidAndersonMarketingServices,Inc.
Pacific Avenue, Suite 14, Palo Alto, California 94301 Telephone 415/555-1212

June 28, 1983

Gerald Bates
es and Company, Inc.
7 Parkwest Plaza, Suite 1222
o Alto, CA 94301

ar Gerry:

re is a quick look at what we have in mind. Using the Sacramento
ore as an example, this is a projection of the sales for the first
x months. These figures are conservative forecasts and follow the
uidelines we discussed yesterday.

BATES AND CO. — SACRAMENTO

Month	Retail Sales	Mail Order Sales	Total
January	82,300	91,100	173,400
February	103,500	146,700	250,200
March	181,450	195,500	376,950
April	156,750	169,900	326,650
May	195,000	213,700	408,700
June	187,500	234,800	422,300
Total	906,500	1,051,700	1,958,200

I have included a quick graph of these figures and, as you can see,
the trend for this location is very good. We will show you simila
figures and trends for all twelve outlets in the full proposal.

We are very excited about the marketing plan we have put
and we are anxious to set up a meeting with you to presen
ideas. I will call your office to schedule something for
the week. I don't mean to rush you, but we need to move
all the obvious reasons.

Regards,

Dave Anderson

- Move from the spreadsheet to another paragraph as easily as hitting the return key.
- You get full use of your printer's special features, graphics characters, enlarged and condensed type styles.
- T/Maker inserts and aligns commas and decimals and justifies text.
- T/Maker does automatic pagination, headers, footers, bold and underscored text.
- This bar chart was made directly from the spreadsheet without re-entering data.

Everything I need in one program.

Because of its flexible approach and powerful built-in features, I can use T/Maker for scores of common applications such as invoicing, proposals, budgets, correspondence, label printing and survey analysis. I decide what I want done and how it should look.

T/Maker is truly integrated —I can have any combination of text, spreadsheets and bar charts all in the same file.

I use T/Maker's full screen editor to enter and format work just as I want it to appear. I enter numerical data the same way—there are no rigid data blocks or cells, no tricky definitions.

T/Maker gives me calculation tools like net present value, projections, growth rates and dozens of others, all built-in. I can use sophisticated word processing tools to automatically add headlines and page bottoms, footnotes, page numbering and many other presentation options.

I also get a number of features that standard word processors and spreadsheets don't offer, like list sorting, tallying or summing over any column or columns in a file.

Not only is T/Maker integrated, but because it generates simple text files, I can also use it to integrate other specialized spreadsheet, word processing and data base programs.

See it for yourself. Ask your dealer for a demonstration and a brochure.

T/Maker III suggested retail price $275. Available for CP/M, CP/M86, MS DOS, and PC DOS.

Simply Great!

T/Maker Company · 2115 Landings Drive
Mountain View, CA 94043 · 415/962-0195

For the dealer nearest you call:

Computer Potentials, Inc.	CA	408/980-9100	1-800-457-4177
DataSolve Corporation	IL	312/943-9141	1-800-621-6872
Lifeboat	NY	212/860-0300	
Software Wholesalers, Inc.	MA	617/587-2904	1-800-633-1000
WESTICO, Inc.	CT	203/853-6880	

T/Maker III is a trademark of T/Maker Company
CP/M and CP/M86 are trademarks of Digital Research, Inc.
MS DOS is a trademark of Microsoft Corporation
PC DOS is a trademark of International Business Machines Corporation

Circle number 16 on the reader service card.

PC, *running in PCDOS*. WordStar, for example, is one of the most popular programs on the IBM PC. Even Digital Research, makers of CP/M, has announced programming languages and Logo to run on PCDOS.

Granted, our readers are users of CP/M *or* PCDOS, or perhaps both (not to mention UCSD Pascal). We intend to continue coverage of the most popular programs on the market, including programs for PCDOS; however, most programs we cover run on *both* systems, and for good reason: what software publisher would risk losing sales by publishing only one version?

No Idle Speculation Here

Although we can't verify the Texas Instruments rumor, we know for a fact that Intel has developed a version of CP/M-86 in ROM called the 80150, or "CP/M-on-a-chip."

What are the advantages of putting CP/M on a chip? First, it lowers the cost of configuring a system, which means that manufacturers can create CP/M-compatible machines for less money, and charge less for them.

Second, the power of a CP/M-based system in such a small package will herald a new era of computer-to-computer communications. Also, truly portable CP/M systems would appear that would not need floppy disks — "bubble memory" (solid state memory that doesn't disappear when you turn the power off) would take its place.

In the workplace, productivity would increase dramatically as various desktop systems are linked in a "local network," sharing printers and large disks, with file compatibility across all computers. CP/M-86 provides a "real-time nucleus" suitable for networking, and with CP/M on a chip a designer could produce less expensive network products.

The development of CP/M-on-a-chip points the way towards future working environments with desktop computers that can perform a variety of tasks at once. This is known as *concurrency*, and it is nearly the opposite of *multi-user* capability. Where multi-user systems slow down as more users run programs, concurrent systems are designed to increase a single user's productivity with simultaneous operations and networking.

More Rumors

As for IBM's new operating system, if there is one: consider (1) that IBM doesn't want to be shackled to new releases from Microsoft, (2) that IBM traditionally led its customers down primrose paths into a garden labyrinth of systems compatible only with IBM's product line, and (3) that IBM has invested in a huge chunk of Intel stock.

What does this mean? We can only guess that IBM, in supporting Intel (which is developing, among other things, CP/M-on-a-chip and RMX-on-a-chip, another real-time nucleus), is indirectly supporting the CP/M world and *at the same time* developing a new operating system to take advantage of the power of the new Intel 286 and 288 chips.

Add to this speculation that Digital Research (makers of CP/M) are conspiring with someone else to produce a "VMS-like" operating system that bridges the gap from CP/M systems to DEC VAX minicomputers (the traditional home of UNIX). Well? No Idle Speculation Here

Of course, IBM has a new computer nicknamed "the peanut" that is supposed to be a MacIntosh killer — a useful portable computer for under $1000. MacIntosh, of course, is the 68000-based portable "mini-Lisa" also to be priced inexpensively. However, it will probably not be as powerful as the $10,000 Lisa unless Lisa drops in price, but it is rumored to be compatible with PCDOS, with CP/M and with UNIX (phew!). Lisa, by the way, was and still is the most advanced new computer system, but do you really care? Rumors are tiring, boring, and sometimes false, but never unexpected.

Reviews?

We've received many letters expressing gratitude for the "hands-on" content that is somehow guiding them through treacherous application software, and thank you for your support!

However, a few have expressed the notion that perhaps we are "hands-only," and since they share a glib "hands-off" attitude toward other magazines, they wanted to see *software reviews* in this magazine.

We're flattered, but we don't consider ourselves to be another computer magazine of software reviews. Granted, reviews in other magazines may be "incredibly dull, boring, indifferent or useless" (they say), but we created *User's Guide* to fulfill another, more painful need for accurate and readable tutorials and user's guides.

We understand the plight of those considering buying computers and software, but the *User's Guide* approach is tailor-made for system users — people who are *already in the grips of despair and are desperate for help.*

User's Guide has the unofficial industry mandate to put systems and software products through *real* paces and file reports on how to use them. With real detailed information and an overall feel for how to use a program, you can make more intelligent buying decisions.

We haven't forsaken buyers and newcomers who have not yet taken the plunge. We are involved in a non-profit organization putting out the first *objective* magazine and catalog of software reviews: the *Whole Earth Software Review* and the *Whole Earth Software Catalog*. The organization is, yes, guessed it — Stewart Brand and the rest of the gang that brought you the *Last Whole Earth Catalog*, the *Next Whole Earth Catalog* and *CoEvolution Quarterly*.

Why say "objective?" We mean untainted with the need to please advertisers or anybody else in the computer industry. Reviews in trade magazines tend to "take a lot into consideration" before publishing a strong review. Not WESC or WESR — the strongest and heartiest will be published without one patronizing thought, or so they (and we) promise...

We've been fans of the Whole Earth crew and their publications for many many years. Who can forget Dan O'Neil's famous Mickey Mouse parody in *CQ*, or the expose of computer hype? Examples of excellent writing and journalism are too numerous to mention. We look forward to seeing the WESR in early 1984 (how fitting) and the WESC in the latter part of '84.

Cache/Q™ by techne´

THE Software Accelerator

—A Most Remarkable Software Product for Your Microcomputer

Microcomputer software is becoming larger and more complex. Performing increasingly sophisticated functions. Demanding more and more of your computer. And longer and longer execution times.

Techne has anticipated this trend. And created a unique software product which can increase the efficiency of your microcomputer. Up to a factor of two. Or three. Or even more.

The name of this remarkable product is Cache/Q.

Cache/Q enhances the operating system of your microcomputer. Extends its capabilities. To a level found in much larger computers. And it does this easily and economically.

What does Cache/Q do? How does it enhance your operating system? In two ways.

Faster Data Access:
First, Cache/Q retains, in the memory of your computer, the most recently accessed disk data. This seemingly simple function has a most profound effect on the operation of any disk-based program. With Cache/Q installed, such a program can run up to two to four times faster. 2 to 4 times faster.

Imagine your precious time saved when a "38-minute job" now takes only 13 minutes. And you perform such jobs many times a week. Or a day.

This dramatic improvement occurs for the following reasons. A sector of data in the memory of your computer requires less than 2 milliseconds (1 millisecond = 1/1000 of a second) to be accessed by your program. That same data on a disk might require up to 300 milliseconds or more to access. And modern data-base programs, accounting programs, and word processing programs typically access thousands of sectors of data during their execution.

Background Printing:
And then there is more. Cache/Q also gives your computer a background print buffering capability. Your programs "print" at the rate of 4000 characters per second. Cache/Q places these characters in buffer memory. Then sends them to your printer at whatever rate they can be accepted. Your program quickly completes its execution, virtually unaffected by the speed of your printer. Allowing you free use of your computer while the previous output is still being printed.

Printing jobs of up to one hour or more can be overlapped with the normal use of your computer (depending upon printer speed and buffer size).

Imagine using your computer while a report or letter is still printing. How many times a day? Think of the time saved. Every day.

As you can see, Cache/Q is a unique product. There is nothing like it on the market. Cache/Q works with your operating system. Increasing the throughput of almost every application you use.

Easily Installed:
Cache/Q is easily installed on the IBM Personal Computer. On any computer using the CP/M 2.2 operating system. And Cache/Q is easily installed on your computer. It can also utilize bank-select memory and I/O-type memory for an even greater increase in efficiency.

Totally Invisible Operation:
Even after Cache/Q is installed in your computer, its presence is totally invisible. To you and your programs. The most recently accessed data is buffered in memory. Automatically. Text output destined for the printer is captured at high speed, buffered in memory, sent to your printer when possible. Automatically. And invisibly.

Cache/Q and "Memory Drives"
Cache/Q's automatic and invisible operation are two major features which distinguish it from much less sophisticated "memory drive" software products. Since Cache/Q is automatic, you need never explicitly copy files from disk to buffer memory, as required by "memory drive" software.

Since Cache/Q is invisible, you need never modify your programs to access an additional "drive," as also required by "memory drive" software.

And, most important of all, you need never worry about a power failure destroying your invaluable data in the "memory drive." Cache/Q writes all modified data onto your disks. Automatically and invisibly.

Cache/Q—The most powerful software product you can buy.

SEE YOUR DEALER ... OR CALL JUDY AT **(800) 835-9595 ... IN CALIF. (800) 835-9594**

IN CANADA CONTACT: CYBERTECH TRADING CORPORATION
Unit 9, 11720 Voyageur Way • Richmond, B.C. V6X 3E1
Telephone: (604) 278-4848 • Toll Free: 1-800-663-9767

Techne Software Corp. • 3685 Mt. Diablo Blvd., Suite 210 • Lafayette, CA 94549 **techne´**

Cache/Q is a trademark of Techne Software Corp. • Cache/Q is a product of Techne Software Corp. • Copyright © 1982 by Techne Software Corp. • CP/M is a trademark of Digital Research • PC-DOS is a trademark of IBM Corp.

Circle number 17 on the reader service card.

Laying a Foundation

Setting Up a Data Base Application

Some expect to be able to handle a complex data base application on a personal computer like an Osborne 01, a Kaypro 2, or even an IBM PC. The essential fact that escapes them is that a data base usually requires more disk storage than is usually provided with your typical personal computer.

It is not surprising to find this misconception everywhere. It starts with microcomputer manufacturers who produce systems with 5.25-inch floppy drives but without a decent "interface" to a hard disk.

Nearly every business and professional microcomputer application could use at least one hard disk, or something with as much if not more capacity. This is *capacity of storage at any one time while using the system.*, not the mythical "unlimited" disk storage capacity of removable media. If the disk is not spinning in the drive, you are not able to use the "extra capacity" when you need it.

If you're planning a data base application, the question to ask is *how much disk storage do I need on-line* while running the data base management system. The answer is usually a ten megabyte Winchester-type hard disk (a good price/performance ratio) or one of the newer, higher capacity hard disks (more expensive but in some cases a requirement).

How Big Is the Data Base?

This is not an easy question to answer. You can use an approximate number of characters per record, which you have to define eventually when you design the structure of the data base. How many characters, spaces, symbols, digits, etc. comprise a complete "record" in your data base?

We use approximately 256 characters, which is a "nice round" number for hexadecimal-based number-crunchers. Two records have 512, four have 1024, which in the jargon is one kilobyte. Four records = one kilobyte?

Not quite. The ratio is less, because there are additional spaces taken up by *instructions to the program* that define where the record starts and ends, where the file itself starts and ends, and how to position to a particular record. In the subscription application presented in this issue, we use approximately one kilobyte to hold every three records.

Three records = one kilobyte. 24-27 records = eight kilobytes, 36-43 records = twelve kilobytes. As the file grows in records, it reaches a point where it grows by another kilobyte; records fill the kilobyte and it reaches another point where it occupies *another* new kilobyte.

You may be able to start the data entry process on floppy disks, but at some point the data will overflow the disk and you'll need separate files. With separate files come the added organizational duties, a file naming and backup scheme, an update process that includes updating the backup files, *ad nauseam.*

Starting data entry on floppies is a recommended method, because you can transfer and merge the files to a master file on a hard disk, and leave behind the backup copies for your archive. Put a date on the data entry disks after using them, and store a copy of each disk in a safe deposit box.

Disk Usage

We have typists enter data into small systems such as the Kaypro 2, the Osborne, the PMC MicroMate, or the Alspa ACI single-drive system.

The single-drive system can hold 596 kilobytes of data on its single-sided eight-inch double-density format (the double-sided can hold 1.2 megabytes, which may be suitable for some data base applications).

The dBASE II program files take up at least 70 kilobytes, the dBASE II "command files" we wrote take up another 20 kilobytes, and other system utilities and related programs combine to take up 250K. "K" stands for "one kilobyte" or 1024 bytes; "M" stands for "one megabyte" or 1024K. A byte, of course, is one character, or space, or digit, or symbol, etc.

Roughly 300K is left for data entry on the Alspa ACI single-drive system, which is the right size for most dBASE II activities.

The Osborne 01 and Kaypro 2 systems both have two floppy drives holding 190K each for a total of 280K.

With 70K required for dBASE II, and another 70K needed for system utilities and command files, we usually reserve one entire floppy disk for the program, and one for the data. This leaves us up to 190K for data, suitable for marathon data entry efforts.

For IBM PC owners with floppy disk drives (holding 320K each), the choices are to (1) use one disk and part of another for data, and the rest of the other for dBASE II program files, or (2) add a hard disk to your PC, or get the IBM XT (hard disk included).

An index file takes up less than one-tenth the space of its associated data base file. If you use one or more index files, reserve space on the same disk as the data base files used with them.

Keep copies of every data base and index file. The size of the media you use for archiving will tell you the maximum size for each data base subfile. For example, if you use single-density eight-inch disks holding 250K each, your largest data base subfile cannot be larger than 250K.

The best and most secure foundation of a data base application is a knowledge of the system's file manipulation commands. You must know how to copy a disk, copy several files, rename files, erase files, find files on multiple copies of disks and organize your disk library. The most advanced data base management system will fail if you use the wrong disk; the most important task ahead of you might be deciding on a disk labeling scheme!

And you thought a *computer* would help *you* organize things.

Letters...

Dear Editors,

I have been receiving your publication since the first issue and I found it very helpful. I really like the level of information that you provide in your articles. It is a good blend between the extremes of articles aimed only at the first time computer users or system level programmers.

I use my Radio Shack Model II in my law practice. When I first bought the computer, Radio Shack didn't have have a word processing program available for the Model II so I decided that I would use CP/M as the operating system and use existing software. The main use is word processing, for which I use the Magic Wand. I also have dBASE II for general record keeping.

I used to think that my situation was rather unusual (only one person in the business understanding anything about computers), but with the boom in the sale of personal computers due to the attractive prices, it must be common for one person at work or at home to be using the computer and not have anyone else around that knows or is interested in it.

I miss not having a more knowledgeable person available to ask for help. Your magazine helps to explain many of the things that I need to know in a manner that makes it easy to understand.

I have put in many hours reading many different publications including the manuals that come with the hardware, software, books and magazines. I am looking to future issues of your *User's Guide* in assisting me to better understand the programs that I am currently using and to assist me in making decisions about purchasing new software.

In particular I want to buy a new version of CP/M, probably CP/M Plus from Radio Shack when it becomes available, a spelling checker program, and possibly a spreadsheet program... At a later date I would like to add a hard disk and another terminal or second computer (if I can have a convenient way for the computers to share data).

Good luck with your publication.

Elise M. Edgell
Orange, CA

Worthwhile

Quite a worthwhile periodical from such a small group of people!

John A. Domansky
Dayton, OH

Editors,

You've done it again! With each issue I have been more and more impressed. The article in this, the fourth, explaining in the detail the floppy disk, is outstanding. As a novice, learning the basics through your magazine has proved a real god-send.

I'm more than happy to renew now. Keep up the good work.

Diane Berge
Seattle, WA

Editors,

... you all have generated impressive reputations in the industry as computer liberators (or computerized liberators) ... I want to make sure I support your work. Thanks for what your doing.

Dan Schaller
Compuserve
Albuquerque, NM

Editors,

Your magazine is the only decent reading material for the professional micro user... I have to teach dBASE II to two assistants and WordStar to three secretaries — can't do it without *User's Guide*. Thank you!

Jeannette McCahon
San Gabriel, CA

WordStar Irritations

One of my biggest irritations with WordStar is that it will not start typing at the line of the page that you line it up

Continued on page 61

An Introduction to dBASE II

The Most Popular Data Base Management Program In Use on CP/M Systems and the IBM PC

by Bob Cowart

© 1983 Bob Cowart

dBASE II from Ashton-Tate has become the de- facto standard for CP/M data base management. Written by Wayne Ratliff (formerly dubbed VULCAN), and introduced 3 years ago in its first incarnation, Ashton-Tate has already sold over 120,000 copies. The obvious conclusion is that dBASE II must be an attractive and useful product.

Many are jumping on the bandwagon — there is a plethora of dBASE II utility programs, books, tutorials, user's groups, training classes and freelance consultants. Ashton-Tate has introduced two additional dBASE II products: a "run-time module" which eliminates the need for the entire dBASE II package when running pre-written applications, and a catalog (*Application Junction*) of application programs which are available from programmers around the world. In addition, the latest revision of dBASE II (version 2.4) includes a HELP command and a debugging function. And according to Ashton-Tate's public relations department, as of Aug 5th, each new dBASE II package will include a 10-part interactive tutorial on disk!

In light of this popularity, it is timely to offer a series of articles explaining what dBASE II is, what it can do, and how the microcomputer user can get started taking advantage of dBASE II's power and flexibility.

In the last issue of *User's Guide*, we discussed some of the elementary issues concerning the structuring of data bases and provided an example client data base. This article expands some of those concepts, and introduces the use of the dBASE II command language for database control.

Why Data Bases?

Many users quickly learn the advantages of word processing and "spreadsheet" financial planning programs, but get confused when it comes to using their computers for organizing and compiling stacks of information. Just how should one go about keeping all those receipts, bills, phone numbers, people, products, or even ideas in order? Computers, with their inexhaustible capacity for drudgery should certainly be the perfect medium for this type of work.

Anyone who has organized (and reorganized) a filing system, or hunted for an address, or prepared a report based on gobs of redundant data can attest to what a waste of time this type of work is. Human-being-hours are better spent in creative and intuitive pursuits than in memorizing lists of things.

Data bases probably made their first appearances on the walls of Cro-Magnon caves to assist in remembering (and avoiding) the migration routes of mastodons. Actually, most of us use at least one "data base" every day, but we don't call it by that name. An encyclopedia, dictionary, phone book or a catalog are common examples. The need for computerization arises only when both the amount of information and/or the speed with which we need to access it increases appreciably.

Enter the computerized "Data Base Management System", or "DBMS." In plain English this means "a way of using modern electronics to aid in the task of storing, updating and then locating all kinds of information efficiently". Sophisticated data base management systems have been operating on large mainframe computers for at least a decade. Offering such sophisticated and flexible systems on a microcomputer can be like squeezing blood from a turnip.

dBASE II is a powerful program that is constantly being improved upon. I believe it is one of the most useful products currently available for micros; however, it is *not* easy for beginners to learn, especially if you want to take full advantage of its capabilities. Actually, dBASE II is two programs in one: A data base filing system, *and* a programming language.

The task of creating and gaining access to dBASE II data bases is relatively simple, and this, to a large degree accounts for the program's success. Simple English words like CREATE, APPEND, LIST and EDIT can be entered from the

dBASE II is a trademark of Ashton-Tate (Culver City, CA).

keyboard for immediate results. However, mastering the subtler points of command syntax and structured programming style may require more time and diligence. The advantages you gain from dBASE II are directly proportional to the time you spend learning its idiosyncrasies.

Creating a Data Base

In the last issue of *User's Guide* we used the command CREATE to create a data base which might be used by a professional contractor. Let's summarize this process, and also learn some commands to control that data base and provide access to your information.

Something everybody can use is an electronic phone-book, or a PEOPLE data base. I use this one everyday. It is easy to update, and I can print out a copy to take with me. It is always in alphabetical order, and is easy to read, unlike my old personal phone book which typically had 10 crossed out phone numbers for each person.

Assuming your system is up and running, and you have DBASE.COM, the necessary assorted support files (refer to your dBASE II manual if in doubt), and some spare room on drive A:, follow along.

Make sure you have followed the dBASE II manual's instructions for running the INSTALL program for your particular screen/keyboard ("terminal"). The INSTALL program is easy to use, and it lists many popular types of terminals and computers with built-in screens and keyboards.

In the following interaction, the words in lower case are the ones *you* type. dBASE II always (except in the case of the North Star Enhanced version) displays its messages in UPPER CASE. Also, Since most terminals and computers are capable of "X-Y cursor positioning," I am assuming you have installed your copy of dBASE II for full-screen editing. If not, most of the examples still apply, though there will be some differences.

With the CP/M A〉 prompt on your screen, type:

```
A〉dbase↵
```

When dBASE II asks for the date, you can either type it in or press RETURN for no date. Typing in a real date can be very useful, because dBASE II uses the date you type in two ways:

1. The date is temporarily stored in a *memory variable* in your computer so that you can use it in a dBASE "program" you may write.

2. It is permanently stored on any data base you happen to update during this session, so that you can find out when the last update occurred.

If you simply press RETURN, the date *00/00/00* is assumed by dBASE II. I sometimes do this when I'm using dBASE II for a quick operation involving no file updates — when I know that I don't have much use for the date information. However, for these examples, type in the correct date.

@ can become HENRI DE TOULOUSE-LAUTREC

IMPOSSIBLE? NOT WITH
SMARTKEY!

SMARTKEY™ is a unique utility that can redefine any ASCII character or function key to become anything you want. For example, "#" becomes "pip b:=a:*.pas[v]". With a single stroke, a key can represent a chosen character or string at the system level or within a program. Instantly. Without rewiring or soldering.

SMARTKEY™ is completely transparent to the user. It resides on the top of memory and intercepts calls to the BIOS, translating system input to whatever you desire. You can even change a key definition while another program, such as WordStar™, is in operation . . . without interruption! It's perfect for programming, data entry or word processing.

"EXCELLENT" InfoWorld
"VERSATILE AND RELIABLE" Lifelines
"WORKS LIKE A CHARM" Microsystems

Think of the acceleration in productivity. Think of the versatility in keyboard layouts. Think of the possibilities. And, best of all, **SMARTKEY™** is only $60.

Ask about **SMARTPRINT™, SMARTSCREEN™, SPOOL™** and other programs.

To order or obtain more information, call or write to:

HERITAGE SOFTWARE, INC.
2130 S. Vermont Ave., Los Angeles, CA 90007/(213) 737-7252

SMARTKEY™ is compatible with all standard versions of CP/M.™
Programs copyrighted by FBN Software.
WordStar™ is a registered trademark of MicroPro, Inc.
CP/M™ is a trademark of Digital Research.

VISA MasterCard

Circle number 10 on the reader service card.

$29.95
full-feature text editor

OMNIWRITER™

The Ideal Tool for Writers, Editors, Typesetters, and Programmers

Multiple Print Options allow you to print any portion of your file. OMNIWRITER also permits you to edit one file while printing another.

Search and Replace allows you to find any character sequence (including "wild cards"). Make as many as 100 different changes in one pass using **Multiple Global Search.**

Multiple File Access allows you to edit up to three files at a time. Select names from a list and merge them with a letter in another file, or move subroutines from one program to another.

Communications: is both batch with XON/XOFF protocol, and interactive for timesharing.

User-Programmable: OMNIWRITER's buffer files store copy and/or sets of commands. Create a library of text-processing programs to fit your own needs. Commands include **Conditional Test and Branch,** which makes complex reformatting simple. Perform **code conversions** from 8080 to 8086 assembly language.

Safety Features: Interrupt any automatic process by simply hitting any key on the keyboard. Also, OMNIWRITER holds all deleted copy in a temporary buffer, from which you can retreive it if needed.

100-page Teaching Manual, 32-page User's Guide, Installation Program included.

☐ 8″ SSSD for CP/M-80 (ver. 1.4, 2.2), MP/M
☐ 5¼″ disk for CP/M-86 (IBM PC and others)
☐ 5¼″ disk for MS-DOS (IBM PC and TI PC)
Other formats available. Inquire.

Send $29.95 plus $3 postage (add $1.95 sales tax in CA) to:

Omni Systems International, 2229 McGee Avenue, Suite H, Berkeley, CA 94703. (415) 644-1896.

Circle number 12 on the reader service card.

```
FIRST:NAME   ¦  LAST:NAME  ¦  COMPANY  ¦ STREET  ¦ CITY  ¦ STATE  ¦ ZIP
-------------¦-------------¦-----------¦---------¦-------¦--------¦----
             ¦             ¦           ¦         ¦       ¦        ¦
```

Figure 1. *A representation on paper of the fields in the PEOPLE data base.*

```
FIRST:NAME   ¦  LAST:NAME  ¦  COMPANY  ¦ STREET  ¦ CITY  ¦ STATE  ¦ ZIP
-------------¦-------------¦-----------¦---------¦-------¦--------¦-----
             ¦             ¦           ¦         ¦       ¦        ¦
John         ¦ Appleseed   ¦Super-seed¦14 E.3rd¦Phila.¦ Pa.    ¦19066
```

Figure 2. *A paper representation of the PEOPLE data base with data.*

You should now have the dBASE II prompt, which is a period. Type the following command to create a new data base:

```
. create↩
```

dBASE II now prompts you for information about the data base. First it asks for the name you want to use. Since this is going to be a people list, let's call it PEOPLE. DBASE II will create a CP/M disk file called PEOPLE.DBF (the extension .DBF stands for "data base file"). You do not have to type in the extension yourself, though you may if you wish.

NOTE

dBASE II expects data files to have the extension ".DBF" but this extension is not required. For example, you could create a data base file called PEOPLE.FUN if you wanted to. However, to open this file for use later, you would have to use its full name including the ".FUN" extension. Also, the command *DISPLAY FILES* (shown later) would not report the existence of PEOPLE.FUN, even though it was on your disk, because the command only looks for files with the ".DBF" extension. Therefore, to avoid confusion, use the ".DBF" extension (which dBASE II automatically applies to the name you give for your data file), unless you have some good reason not to.

Defining the Structure

After creating the name of the data base, you must *define its structure*. The following shows what you must type to define the structure. If you get the message BAD NAME FIELD or other error message, check to see that you have put the commas in the right place, and typed the field names properly (including the colons):

```
ENTER FILENAME: people↩

ENTER RECORD STRUCTURE AS FOLLOWS:

FIELD       NAME,TYPE,WIDTH,DECIMAL PLACES
001         first:name,c,10↩
002         last:name,c,15↩
003         company,c,20↩
004         street,c,20↩
005         city,c,12↩
006         state,c,2↩
007         zip,c,5↩
008         h:phone,c,12↩
009         w:phone,c,12↩
010         notes,c,50↩
011         ↩

INPUT DATA NOW (Y/N)? n↩
```

Planning and defining the structure of a data base is possibly the most important and most challenging task, as well as the most confusing aspect of using dBASE II. Here's a simple explanation of the structure you just defined.

If you can visualize the "paper parallel" of the structure you created, you should be able to understand the process. Think of each *field* in the data base as a separate vertical column on a piece of lined paper, as shown in figure 1.

When you are ready to enter names and information into the data base, each "record" (each name) will occupy (conceptually) one horizontal line on the page, as shown in figure 2.

Once you think of your data base in these terms, then the process of creating the structure will become more automatic. Think of the structure as a blank index card with underlined areas to fill in names and addresses. With an index card data base, you have to write on each card; with dBASE II, the "cards" are set up automatically and you only have to type the information. Don't confuse the *structure* with the actual *data*.

Field Types

There are basically three kinds of data: *characters* (letters, symbols, spaces, digits, etc.), *numbers* (actual values to be used in calculations), and *true/false* answers to specific questions, which are called *logical* answers.

The *c* after each field name in the above structure definition defines the field as one that will receive characters (or, if you use BASIC, "string data"). Every field in our data base will receive character data, which is usually the case with most data bases.

Although the phone number is obviously all numbers, we defined the field as a character field because digits can be considered characters. Why use a character field for the phone number? Unless you need to perform arithmetic on that particular field (like finding the SUM of all your friend's phone numbers), both you and dBASE II will find it easier to use character fields. Number fields should only be used when they are necessary for calculations.

Field Length

The *length* of each field is another very important detail that can also be a stumbling block. You can easily paint yourself into a corner by using inappropriate lengths.

If you define a field too large, it will waste disk space since blank spaces are saved on disk along with your data. If you define a field too small, it will force you to abbreviate the data. Phone numbers are easy, but addresses and names take a little trial and error. The numbers we used work fine for these examples, but you should think about lengths before you define a data base for a real application.

The maximum length per field in dBASE II is 256 spaces, but each record (horizontal line in figure 2) may not exceed 1000 characters.

Verifying the Structure

It's always a good idea to verify what you have done. To check whether dBASE actually recorded the structure properly, type the following USE and DISPLAY STRUCTURE commands:

```
. use people↵
. display structure↵

STRUCTURE FOR FILE: A:PEOPLE   .DBF
NUMBER OF RECORDS: 00000
DATE OF LAST UPDATE: 00/00/00
PRIMARY USE DATABASE

FLD         NAME         TYPE    WIDTH  DEC
001         FIRST:NAME   C       010
002         LAST:NAME    C       015
003         COMPANY      C       020
004         STREET       C       020
005         CITY         C       012
006         STATE        C       002
007         ZIP          C       005
008         H:PHONE      C       012
009         W:PHONE      C       012
010         NOTES        C       050
** TOTAL **                      00159
```

FILEBASE™

A menu-driven interactive processor for BASIC type files. Create, append, merge, sort and print files. Rearrange, remove and add fields. Update records by random access. Create and process files for MailMerge™. Interface FILEBASE™ created files with BASIC programs.

"...a convenient utility package for manipulating files and their associated fields."

"...quite easy to use." "...documentation is programmed..."

CP/M® Review, Jul/Aug 83

(reprints available from EWDP)

"...user does not need to know the number of fields or their sequence in the record..."

Microsystems, p122, Aug, 83

Select/exclude "...by interactive examination of records in real time."

Lifelines™ p36, Jun 83

Select/Exclude by 5 Methods

1. Test of contents [LT, GT, EQ,LE, GE,BT]
2. List of 300 items [Jones, Smith, etc.]
3. Range of record numbers [27 - 86]
4. List of 1000 record numbers [5,7,25,37]
5. Look at records - decide by keystroke

Send excluded records to second file

Special Sort Features

Sort on last name even if embedded in full name
Sort on ZIP even if part of city, state, ZIP field

Print

Envelopes - Labels - Line Listings
Titles - Headings - Total Numeric Fields

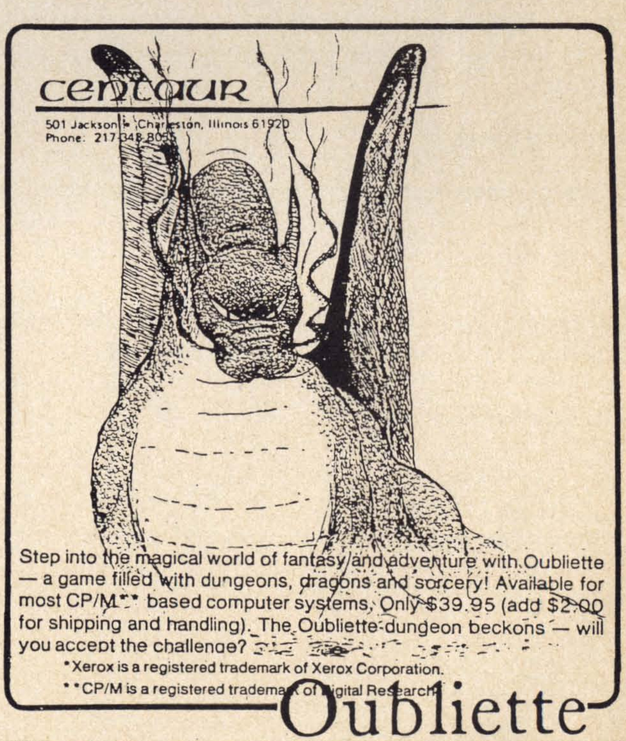
Dealers — See us at SOFTCON Space 2022

Z80, 8080, 64K, CP/M®
(Special version for Osborne 1™ available)
$75.00 - Check, AMEX, VISA, Mastercard

EWDP Software, Inc.
P.O. Box 40283
Indianapolis, Indiana 46240
(317) 872-8799

CP/M® is a registered trademark of Digital Research, Inc.
MailMerge™ and DATASTAR™ are trademarks of Micropro®.
Osborne 1™ is a trademark of the Osborne Computer Corp.

Circle number 7 on the reader service card.



centaur

501 Jackson • Charleston, Illinois 61920
Phone: 217-348-8055

Step into the magical world of fantasy and adventure with Oubliette — a game filled with dungeons, dragons and sorcery! Available for most CP/M** based computer systems, Only $39.95 (add $2.00 for shipping and handling). The Oubliette dungeon beckons — will you accept the challenge?

*Xerox is a registered trademark of Xerox Corporation.
**CP/M is a registered trademark of Digital Research.

Oubliette

Circle number 2 on the reader service card.

Other information is displayed with the structure: the date that your data base was last updated and the number of records or entries you have stored thus far. If everything looks okay, then type the DISPLAY STATUS command (dBASE II version 2.4):

```
. display status↵
DATABASE SELECTED - A:PEOPLE .DBF
PRIMARY USE DATABASE

WAITING  ↵   — Press RETURN.

TODAY'S DATE       — 00/00/00
DEFAULT DISK DRIVE — A:

   ALTERNATE  — OFF    BELL      — ON
   CARRY      — OFF    COLON     — ON
   CONFIRM    — OFF    CONSOLE   — ON
   DEBUG      — OFF    DELETE    — OFF
   ECHO       — OFF    EJECT     — ON
   ESCAPE     — ON     EXACT     — OFF
   INTENSITY  — ON     LINKAGE   — OFF
   PRINT      — OFF    RAW       — OFF
   STEP       — OFF    TALK      — ON
```

The *DISPLAY STATUS* command is useful for determining the state of dBASE II features. Most of these can be changed using the SET command (see the dBASE II manual). The *DISPLAY STATUS* command also tells you the name of the selected data base, and the fact that it is the "primary use" data file.

Keep the *DISPLAY STATUS* command in mind for future use. Sometimes when things don't seem to be working right in dBASE II this information will give you a clue. The meanings and applications of the other parameters are not necessary to know at this time (they will be covered in a future article).

Entering Data

To begin entering real data, type the APPEND command:

```
. append↵
```

Assuming that dBASE II is correctly installed for your computer system, you should see something like this:

```
RECORD# 00001

FIRST:NAME     :          :
LAST:NAME      :               :
COMPANY        :                        :
STREET         :                        :
CITY           :               :
STATE          :  :
ZIP            :     :
H:PHONE        :            :
W:PHONE        :            :
NOTES          :
```

Now you can actually go ahead and enter your names and addresses. Use UPPER CASE letters if you want to be able to find data easily without having to guess if the data is typed in UPPER or Upper/lower case.

Don't bother entering them in alphabetical order, since we can use dBASE II to alphabetize the file at a later time. (After all, that's the kind of thing that computers were invented for.)

Press RETURN at the end of each line, unless the data completely fills the line — then dBASE II automatically puts you on the next line. When you type one full record (one name, address, etc.), press RETURN after typing the last field. When you press the RETURN key on the last line, dBASE II brings up another blank "form" for the next record.

Making Changes While Entering Data

If your version of dBASE II is installed for full screen editing, you can experiment with moving the cursor around from line to line. To move the cursor down one line, hold down your Control key and type an *X* The combination of Control and *X* is called Control-*X* and is also shown, in this tutorial and in other article, as ↑*X*.

The cursor movements are useful in case you have made a mistake on a previous line — you can move to that line and type over the old data with new data. If you are a WordStar® user, you should immediately recognize these commands which perform a similar function in WordStar:

↑*X*	Move the cursor one field ahead (one line down).
↑*E*	Move the cursor one field back (one line up).
↑*D*	Move the cursor one space ahead within a field (on a line).
↑*S*	Move the cursor one space back within a field (on a line).
↑*Y*	Erase the field's contents (erase the data on one line).

Sample Data

While you are experimenting with typing names, addresses and phone numbers, type the following fictitious names somewhere in the middle of your batch. We will use them later in the tutorial (and eventually eliminate them):

```
RECORD# 00001

FIRST:NAME     :WOLFGANG↵
LAST:NAME      :MOZART↵
COMPANY        :CHURCH OF AUSTRIA↵
STREET         :135 G. CLEF COURT↵
CITY           :SALZBURG↵
STATE          :CA↵
ZIP            :13579↵
H:PHONE        :800-555-1212↵
W:PHONE        :213-123-4567↵
NOTES          :↵
```

```
FIRST:NAME    :PAUL ↵
LAST:NAME     :HINDEMITH ↵
COMPANY       :WORLD HARMONY, INC. ↵
STREET        :2 RONDO RD.SUITE 122↵
CITY          :ACAPPELLA ↵
STATE         :VT ↵
ZIP           :02468 ↵
H:PHONE       :802-555-1212 ↵
W:PHONE       :802-024-6810 ↵
NOTES         :↵

FIRST:NAME    :JACOPO ↵
LAST:NAME     :PERI ↵
COMPANY       :EARLY OPERA LTD. ↵
STREET        :1597 DAPHNE DRIVE ↵
CITY          :FLORENCE ↵
STATE         :IA ↵
ZIP           :98765 ↵
H:PHONE       :512-555-1212 ↵
W:PHONE       :512-987-6543 ↵
NOTES         :↵
```

As you press RETURN to end a record, dBASE II displays the next blank record for you to fill in. Once you have typed in all of your names and addresses, stop the APPEND process of entering data by pressing RETURN when the cursor is on the first position of the FIRST:NAME field of a new blank record.

Closing/Opening the File

Congratulations. You now have a soon-to-be-useful data base. In fact, by learning a few additional commands, you should be able to access the file and make changes to the data without knowing anything more about dBASE II.

However, before continuing with experiments, make sure that the file is closed properly and all data currently stored in your computer's memory is written out to disk. To do this, type the following USE command:

```
.use↵        — close the data base
```

To start again with the PEOPLE data base, type the USE command again:

```
.use people↵     — re-open and use PEOPLE.DBF
```

The Record Pointer

You may have noticed that during the append process, a five-digit number appeared at the top of the screen. This is the *record number* used by dBASE II to keep track of where you are in the data base. Just as your cursor can only be on one line at a time, dBASE II can only point to one record at a time.

Every person in your data base occupies one record, and every record has a unique number. The current record number can always be determined by typing a question mark, a space, and the # symbol:

```
. ? #↵
```

The # symbol is the *record number function*, and the ? is shorthand for "Display the value of...". dBASE II should respond by displaying a five-digit record number.

This procedure is often useful for extracting information from your data base. It is through the manipulation and adjustment of this record number, either by you, or automatically by dBASE II, that allows you to retrieve data.

However, the current record doesn't "mean anything" in relation to your data. There are a variety of ways of look up data without knowing the actual record number, as you shall see in this and in later installments. dBASE II's response to your last inquiry was probably the last record of your data base. You can always determine the length of your data base with the following two commands:

```
. goto bottom↵
. ? #↵
 163
```

The example shows that I have 163 records in my data base.

The record number pointer can be moved to the top of the data base by typing:

```
. goto top↵
```

You can move forward or backward any number of records using the SKIP command. Here are some examples:

```
. skip↵            — move ahead 1 record.

. skip + 5↵        — move ahead 5 records.

. skip - 1↵        — move backwards 1
                   record.
```

Displaying Data: LIST and DISPLAY

With the period (.) prompt back on your screen, try a few commands:

```
. list↵
```

This command should display a lot of information, scrolling up and off your screen. The information should look familiar to you, though not very orderly. The LIST command starts from the beginning of your data base and displays everything you entered so far, in the most concise format, and in as few lines as possible.

You can use LIST to display the information in a more orderly fashion. Try these to see a more coherent listing:

```
. list first:name,last:name↩
```

This example lists the first and last names of each person in the data base, along with the record number. Try another LIST command:

```
. list first:name +' '+ last:name↩
```

This example adds a few spaces between the two fields during the listing process. Notice that once again the record number makes an appearance. Try another:

```
. list off first:name,last:name,h:phone↩
```

This example displays the first name, last name, and phone number fields. The record number is *not* displayed because we used the *OFF* keyword with the LIST command.

If you have a printer connected to your system, and it is turned on, set to "on line" and working properly, type this command:

```
. SET PRINT ON↩
```

Now repeat one of the LIST commands from the above examples, and the printer should print the same thing you see on your screen. This is the easiest way to print a listing of your data base.

So far it looks as though you paid several hundred dollars for a program that makes lists. What about finding a particular name? This can be done in many ways, but the DISPLAY command is a good place to start. Try the following exercises. Don't forget the colons and the quotes, and use UPPER case when indicated.

```
. goto top↩
. ? #↩
. display ↩
. display first:name,last:name↩
. display record 5↩
. display next 5↩
. display next 5 first:name,last:name↩
. display for first:name= 'WOLFGANG'↩
. display for first:name= 'wolfgang'↩
. display first:name,last:name, ;
  city for first:name= 'WOLFGANG'↩
. display first:name,last:name,city ;
  for first:name= 'WOLF'↩
. display first:name,last:name,h:phone ;
  for last:name= 'S'↩
```

(You can break a command line into two lines by using a semicolon on the first line.)

There are several things you should notice from these examples:

- Like the LIST command, DISPLAY can display selected fields if you specify them; otherwise, it displays all fields in the record.
- If you know the number of the record you want to see, you can specify it for display.
- Upper and lower case are important. dBASE II couldn't locate *wolfgang* because we entered *WOLFGANG* into the data base.
- However, using *WOLF* locates the record with *WOLFGANG* in it, and locates any other records with *WOLF* in the first 4 letters of his/her first name.
- In the last example, *S* should display all records whose last names begin with an "S."
- Notice that the DISPLAY command by itself shows only one record. When you use the *FOR* phrase, dBASE II goes to the top of the data base and displays each record which meets the criteria stipulated in the *FOR* phrase.

The syntax of the dBASE II "command language" makes sense once you get used to it. Try some other commands on your own. The worst thing that can happen is that dBASE II will give you an error message. For example, suppose I type:

```
. disp last:name for first:name=john↩
```

dBASE II responds with the following error message:

```
*** SYNTAX ERROR ***

disp last:name for ?
disp last:name for first:name=JOHN
CORRECT AND RETRY (Y/N)?
```

At this point, you have a choice: you can correct the mistake in the command line by typing *Y* to this question, or you can completely re-type the command by typing *N* to this question and returning to the period (.) prompt.

Let's try correcting the line. The question mark appears above the word that dBASE II could not understand. It should have understood the field name *first:name*, but it did not understand the entire phrase *first:name=JOHN.*

Why not? What happened was, we forgot to use quotes around the word *JOHN*. In fact, it automatically capitalized the word from *john* to *JOHN* in its attempt to understand it. The question mark tells us how far dBASE II got before becoming confused with our syntax. From this "hint" we can guess at the faux pas. Always check for such a clue whenever you cause a ***SYNTAX ERROR*** message.

To correct the line, type *Y* to the question:

```
disp last:name for ?
disp last:name for first:name=JOHN
CORRECT AND RETRY (Y/N)? Y

CHANGE FROM :JOHN↩
CHANGE TO :'JOHN'↩

disp last:name for first:name='JOHN'
```

```
MORE CORRECTIONS (Y/N)? N

00009 DVORAK
00032 CARMICHAEL
00035 COWART
00081 FYFE
00131 CARMICHAEL
00141 THORNTON
```

With the knowledge of record numbers and how to locate specific records, you can easily learn the technique for *editing* them. Editing is necessary in case you need to change some of the data in a particular record. The procedure is simple. There are two ways to do it:

```
. edit↵
ENTER RECORD # : 10↵
```

In this example, you type the EDIT command and enter the record number, followed by RETURN.

The alternate method is to specify the record number with the EDIT command like this:

```
. edit 10↵
```

In either case, you will be presented with a screenful of information similar to what you saw while using the APPEND command. Now you can change the data in the record.

While in EDIT mode you can use the familiar cursor movement commands (Control-*X* to move down, Control-*E* to move up, etc.) to move around the record, and you can type over old data with new data. You can also perform other editing functions. Here's a summary:

↑*X*	Move the cursor one field forward (one line down).
↑*E*	Move the cursor one field back (one line up).
↑*D*	Move the cursor one space forward within a field.
↑*S*	Move the cursor one space back within a field.
↑*Y*	Erase the field's contents.
↑*V*	Turn on or off the insert mode to insert characters.
↑*G*	Erase one character at the cursor position.
↑*W*	Save the changes you have made.
↑*Q*	Ignore the changes you made (the record is unchanged).
↑*U*	Mark the record for later deletion from the data base.
↑*C*	Move forward one record.
↑*R*	Move backward one record.

These are familiar to WordStar users; however, ↑*Q* should not be used unless you are "quitting" from a record without making any changes to it (this is not the same as the WordStar ↑*Q* command).

For an example of editing, suppose you want to find Wolfgang Mozart's record in order to change his address (he did travel a bit). Can you think of the easy way to do it? Try it yourself, then if you need help, try this:

```
. display # for last:name= 'MOZART'↵
```

A record number should appear at this point. Suppose the number was 20. Typing the following will bring up the record in the edit mode:

```
. edit 20↵
```

However, since the record number pointer was already positioned on MOZART's record by the DISPLAY command, you could have simply typed:

```
. edit #↵
```

Actually, now that you have learned the hard way, there is an even faster method of achieving all of this using another command — LOCATE. This command is specifically designed to place the pointer just where you want it. Using the same example:

```
. locate for last:name= 'MOZART'↵
. edit #↵
```

Once you have the record on the screen, use the Control commands to move the cursor to the field you want to change. Type over the old address with a new one. When finished, type ↑*W* (Control-*W*) to save the changes.

By the way, if you try to locate a name which is not in your data base, or you accidentally misspell the name, dBASE II will respond as follows:

```
END OF FILE ENCOUNTERED.
```

Checking the record number with the *? #*↵ command will verify that the pointer is actually on the last record of the data base. dBASE II searched the data base from top to bottom and did not find a record containing the information requested, so the pointer was left sitting at the bottom of the data base.

Next Time

In the next issue, we will explore the areas of *indexing* and *sorting* a data base. Included will be a convenient program for keeping your People list updated as well as for looking up people based on their first name, last name, and company. In the meantime, continue updating your list, and be sure to keep a regularly updated backup copy!

Issue No. 1 Sold Out!

Issue no. 1 of *User's Guide* is sold out. We are planning a revised edition with additional material for a cover price of $5. If you wish to be notified of availability, send your name and address to We Want No. 1 Dept., User's Guide, PO Box 3050, Stanford, CA 94305.

Using CP/M Plus, Part One: Commands and File Utilities

From System Start-up and Command Line Editing To File Attributes, Passwords and Utilities

by Tony Bove & Cheryl Rhodes with help from Steve Rosenthal & Kelly Smith

CP/M and CP/M Plus are trademarks of Digital Research, Pacific Grove, CA.
WordStar is a trade mark of MicroPro Intl., San Rafael, CA.

CP/M Plus® is a major update of Digital Research's CP/M® operating system for eight-bit computers. It is also referred to as CP/M 3.0.

This tutorial leads you gently into CP/M Plus and its operating utilities. We used several CP/M Plus systems in our examples: the PMC MicroMate (described in a separate article in this issue), the Osborne Executive, and the Bullet single-board computer from Wave Mate, Inc.

All CP/M Plus systems are supplied with Digital Research manuals, and most are supplied with documentation on specific deviations from standard "vanilla" CP/M Plus. We also assume you have access to a reference guide and summary of CP/M Plus such as "Reference & Summary: CP/M Plus Operating Tools" in *User's Guide* vol. 1 no. 4.

CP/M Plus is produced by Digital Research, Inc. (the producers of the original CP/M), and available bundled with many new personal computers such as the PMC MicroMate and the Osborne Executive. CP/M Plus is available for the Radio Shack Model 4, for the Apple IIe (bundled with an add-on circuit board to convert an Apple to use CP/M), and other popular systems, and it is also rumored to be the system for an as yet unannounced IBM personal computer.

Symbols and Typeface Conventions

In examples we use *THIS TYPEFACE* to show commands and expressions you type at your keyboard, using all *UPPER CASE* letters for command words and program names used as commands, and *lower case* letters for values you supply when typing the commands (you of course can use either upper or lower or both when typing commands and names of programs).

We also use This typeface to show messages displayed by the system or by programs. In addition, we use these symbols:

↩ The RETURN key (CR or ENTER on some keyboards). This key is used at the end of commands and other typing.

↑ The Control key (CTRL, CNTL or ALT on some keyboards). This key is used in conjunction with other keys to perform *Control key commands*, used mostly to edit CP/M command lines, to "wake up" CP/M with Control-*C* (Control and *C* keys at the same time), or to perform functions in programs like WordStar®.

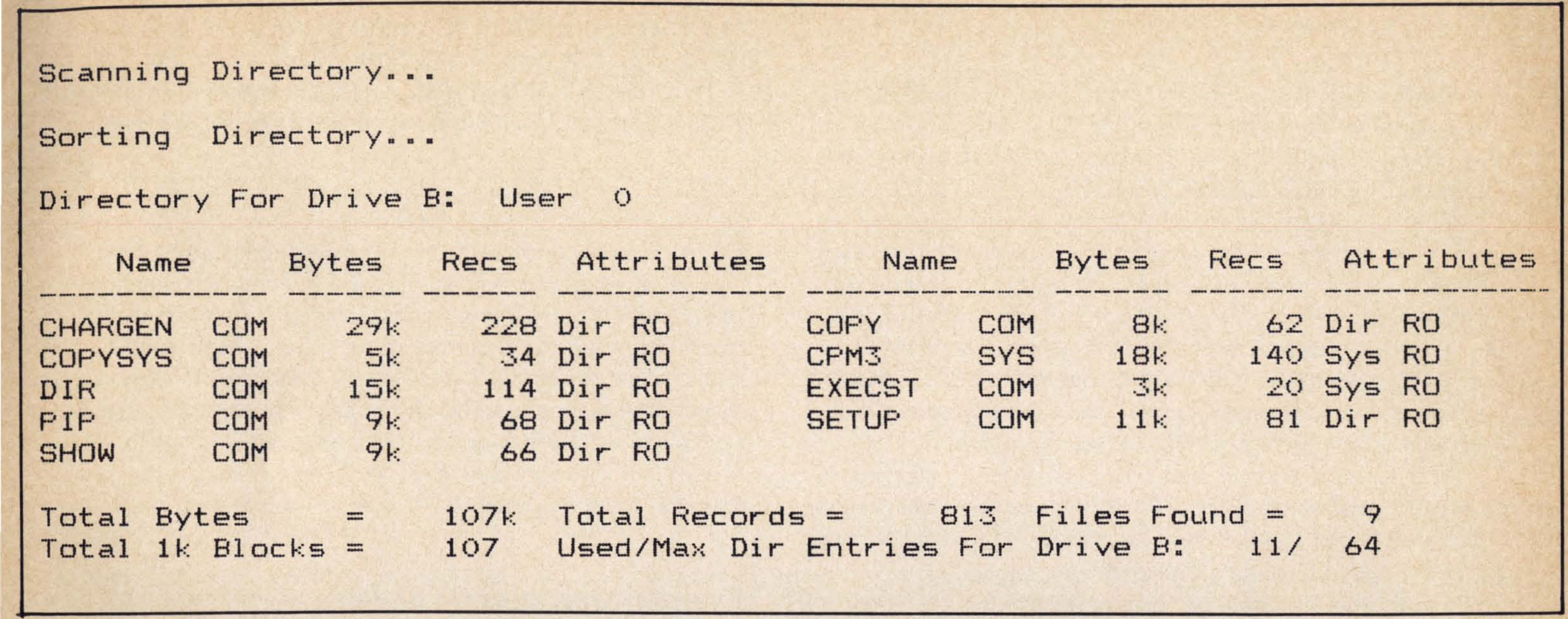
Starting Up or Starting Over

As with regular CP/M, you start or re-start a CP/M Plus system by pressing the computer's reset button. Many computers are supplied with an automatic self-testing program that checks the system's memory before starting CP/M Plus. CP/M Plus takes a few seconds to perform system "housekeeping," then it displays the familiar A⟩ *prompt symbol* signifying the CP/M command line.

To get the feel of this new system, type the following command as shown (including the square brackets and space between *DIR* and *[FULL]*):

```
A⟩DIR [FULL] ↩
```

You should see a directory display similar to the one



```
Scanning Directory...

Sorting  Directory...

Directory For Drive B:  User  0

    Name       Bytes   Recs   Attributes        Name       Bytes   Recs   Attributes
------------ ------- ------ ------------    ------------ ------- ------- ------------
CHARGEN  COM     29k     228 Dir  RO        COPY     COM      8k      62 Dir  RO
COPYSYS  COM      5k      34 Dir  RO        CPM3     SYS     18k     140 Sys  RO
DIR      COM     15k     114 Dir  RO        EXECST   COM      3k      20 Sys  RO
PIP      COM      9k      68 Dir  RO        SETUP    COM     11k      81 Dir  RO
SHOW     COM      9k      66 Dir  RO

Total Bytes      =     107k  Total Records =      813  Files Found =     9
Total 1k Blocks =      107   Used/Max Dir Entries For Drive B:     11/  64
```

Figure 1. *Directory display using DIR and the [FULL] option.*

shown in figure 1. Most commands are typed in this sequence — a command word, a space, perhaps the name of a file, another space, and perhaps an *option word* in square brackets. Commands are sent to the computer when you press the RETURN key.

Control-Key Commands

The Control key is used in conjunction with another key to move the cursor, change a command, cause the system to restart or re-awaken a disk drive, and do other functions. You hold down Control while typing another key (usually a letter).

Moving the Cursor and Editing

If you're using the banked version of CP/M Plus (more than 64K memory — usually the 128K version), type a Control-*W* and CP/M repeats your *DIR [FULL]* command:

— type Control-*W*
A)DIR [FULL]

Press RETURN, and the command executes. At the A) prompt, type Control-*W* to repeat the command. Then type Control-*X* to erase the command line. Type RETURN a few times, and type Control-*W* again. The command should repeat again.

Using Control-*A* commands, move the cursor back to the space between DIR and [FULL]. Type Control-*R* to erase the [FULL] option, and press RETURN to execute DIR without the option. This practice should give you a feel for using the commands.

To move the cursor over the command line, use ↑*A* to move left, and ↑*F* to move right (↑ stands for the Control key). Use ↑*B* to move quickly from one end of the command line to the other without changing the command.

The banked version of CP/M Plus also offers the ↑*G* command, which deletes one character at the current cursor position (if the cursor is in the middle of a command line). The ↑*U* command cancels the command line and moves the cursor to the next line, leaving a # symbol on the command line. You can then use ↑*W* to recall the partial command line (up to the # symbol).

Here's a quick summary of editing commands:

↑*A*	Move the cursor one character to the left. Banked system only.
↑*B*	Move the cursor from one end of the command line to the other without changing the command. Banked system only.
↑*F*	Move the cursor one character to the right. Banked system only.
↑*G*	Delete one character at the current cursor position if in the middle of a line. Banked system only.
↑*H*	Delete one character to the left of the cursor.
↑*K*	Delete character at cursor and all characters to the right of the cursor.
↑*R*	Retype the characters to the left of the cursor on a new line. This changes the command line buffer by erasing unwanted characters to the right of the cursor. Use this to repeat the command line without rubbed-out (deleted) characters.
↑*U*	Cancel the command line and move the cursor to the next line, leaving a # symbol on the command line. You can then use ↑*W* to recall the partial command line (up to the # symbol).
↑*W*	Recall the previous command line if current line is empty; otherwise move the cursor to the end of the line. ↑*J*, ↑*M*, ↑*R*, ↑*U* and RETURN update the command line for recall with ↑*W*. Banked system only.
↑*X*	Delete all characters to the left of the cursor.

System Functions

Some Control-key commands perform system functions like starting the printer, stopping a scrolling display, or switching disk directories. The most often used commands are usually Control-*C* and Control-*P*.

Control-*C* stops whatever program is executing and returns you to the CP/M command line. Control-*C* is also used whenever you change a disk, in order to switch disk directories.

Control-*P* sends to the printer everything you type and everything subsequently displayed; your printer must already be turned on and properly connected to the computer (connect everything before turning on your equipment).

The Control-*S* command stops displays to give you a chance to read them. Since CP/M Plus has been enhanced to provide a "page" mode where scrolling stops for each screen, Control-*S* is not needed for most programs.

The system functions are summarized:

↑*C*	Stops a program's execution, or when entered at the system prompt, re-starts the system and reads a new disk's directory. Use this command after changing a disk in any drive, unless you are instructed to change disks by a program.
↑*E*	Extends a command to the next line by forcing a physical "carriage return" without sending the entire command line to the system.
↑*P*	Sends your display and keyboard input to the printer (echoes console input and output to the list device).
↑*Q*	Re-starts screen scrolling after a ↑*S*.
↑*S*	Stops screen scrolling until a ↑*Q* is typed.

Formatting and Copying

Most computer manufacturers supply their own version of the COPYDISK or COPY program. You should see such a program in your directory listing, followed by the letters "COM." The name COPY and the letters "COM" form the entire filename *COPY.COM.* A file of type "COM" usually contains a program you can run by typing its name.

For example, you type *COPY* (or *COPYDISK* or something like that, depending on your computer) to make a copy of a disk. The program asks for the *source* of the copy, and you respond with the drive (A, B, etc.) holding the disk to be copied. Some copy programs assume you are using drive A as the source, and you only need to respond if you are changing the source to another drive.

The program also asks for the *destination*, which is the drive holding the new disk to receive the copied files. Again, some programs assume drive B to be the destination; they assume you're copying from drive A to drive B. You must have a *formatted* disk in the destination drive — that is, a disk formatted for your computer.

Nearly every computer has a different disk format, and therefore a different disk formatting and copying program. You have to learn how to *format a disk* (prepare one for use) and make a copy of it, before moving on. Consult your manual — look for information on formatting and copying disks.

Don't continue using the only copy of a master disk. Make at least two copies, and retire the master disk for safekeeping.

NOTE

CP/M Plus has more utilities that occupy disk space (called "transient programs") than CP/M 2.2. Some are expanded versions of CP/M 2.2 commands, and you may not be used to reserving space for them on system disks. Each time you make a new system disk you have to remember to copy the files corresponding to these commands. Use a track-by-track *COPYDISK* or *COPY* program if you have one, to copy an *entire* system disk to a new disk, then delete files on the new disk that are not needed.

Everyday Operations With Files

CP/M Plus provides several utilities for managing files and disks. You can set up a disk to have a disk password to prevent unauthorized use, and you can assign passwords to individual files. The time and date can be set for each file and changed automatically when you update the file, so that you can tell which version of the file is the latest version.

Built-in DIR Displays

The *DIR* command displays the names of the files on the selected disk. *DIR* does not display files that are "hidden" by means of the SYS attribute — for a display of "system" files, use *DIRS* (an abbreviation of *DIRSYS*). You must specify a disk drive to see the directory of a drive other than the current drive:

```
A>DIR B:↵
```

You can select filenames that match a *filematch* consisting of the asterisk (*) and question-mark (?) symbols. A *filematch* uses * to match any number of characters in a filename, and ? to match any one character in a filename. Use a *filematch* to find several files at once. You can precede *filematch* with a *d:* drive specifier. Example:

```
A>DIR B:CHAP?.*↵
```

— This example displays all DIR attribute files on drive B (user area 0) whose filename consists of the letters CHAP and one other number or letter (or symbol), followed by any extension.

Expanded DIR Utility

CP/M Plus comes with an extra DIR.COM utility program loaded with options. You specify an option after the word *DIR* and enclosed in square brackets (you can omit the closing bracket in most cases):

```
A>DIR [FULL⏎
```

You've seen this command, which lists every file in the current user area in alphabetical order, with all the information pertaining to each file. However, it *does not* display *system files* — files marked with the "system" attribute that make their names invisible to ordinary DIR commands.

To display system files, use the built-in DIRS command, or the *[SYS]* option with the DIR command:

```
A>DIR [SYS] ⏎
```

The default for the expanded directory command is a listing in alphabetical order, with added information depending on any options you specify. The options replace most of the old STAT command features available in CP/M 1.4 and 2.2.

The *[FULL]* option gives you a full display of everything, except system files, in the current user area. The following provides a full display of non-system files in *all* user areas of the current drive:

```
A>DIR [USER=ALL] ⏎
```

User areas are not that useful in systems with floppy disks as the only disk storage, but are very convenient in systems with a hard disk. If "all" user areas are too many, the DIR utility lets you specify which user areas to display:

```
A>DIR [USER=2] ⏎
A>DIR [USER=(1,2,5,7)] ⏎
```

The first example displays the files in user area 2 only; the second example displays the files in user areas 1, 2, 5 and 7.

To get a full display of every file in the system, try this command:

```
A>DIR [USER=ALL,DRIVE=ALL] ⏎
```

— This example displays the filenames of all files in all user areas (0-15) on all on-line disk drives. This may take some time. If your system is set in "page" mode, the system will stop at each screenful and wait for you to press RETURN before continuing the display.

If you want to be more selective about what you display, use the *[EXCLUDE]* option to display files that *do not* match the filename or filename match you supply:

```
A>DIR [EXCLUDE] *.DAT⏎
```

— This example displays all the files in user area 0 on drive A that do *not* have the ".DAT" extension.

If the full DIR display is too much information, and you only want to know the sizes of files, use the *[SIZE]* option. The *[DATE]* option will not work unless you have already set up date and time stamping, discussed later. Also, if you have not yet set up a disk password and file passwords, the full display will present abbreviated information in two columns of filenames; with the disk and file passwords set, a full display

CUTE ©
by
User Friendly Software®

Client writeUp sysTEm
for the accounting professional

CUTE© — easily installed, easy to use — brings your practice up to a high level of efficiency. Self-verifying data entry makes trial balance unnecessary, and you can trust non-professional assistants to process clients with this User-Friendly Software familiar posting and entry technique.

Multi-user system includes General Ledger, verified bank reconciliation, comparatives, budget and post-facto payroll reporting — in standard charts and reports which are easily customized to your own formats.

- Self-prompting — menu-driven
- Standard operating system: (C/PM, M/PM) runs on most micros
- Written in PL/1.

To find out how easily and inexpensively you can streamline your practice with CUTE, call or write:

User Friendly Software ®

1625 South 21st Avenue
Hollywood, Florida 33020
Dade (305) 949-8319
Broward (305) 925-8194

Circle number 18 on the reader service card.

LYNC

COMMUNICATIONS SOFTWARE

WHY BUY 3 PROGRAMS?
LYNC HAS ALL 3 IN ONE!

1) FILE TRANSFER 2) REMOTE OPERATION
3) COMMUNICATE WITH TIME-SHARE

Easy set up on most systems.
Menu driven install program.
Command Mode of operation
Help Screen for Lync & Term

Z80 based CP/M $155
PC DOS $155 TRSDOS $75

ISA

We're Not the Biggest, Just the Best.

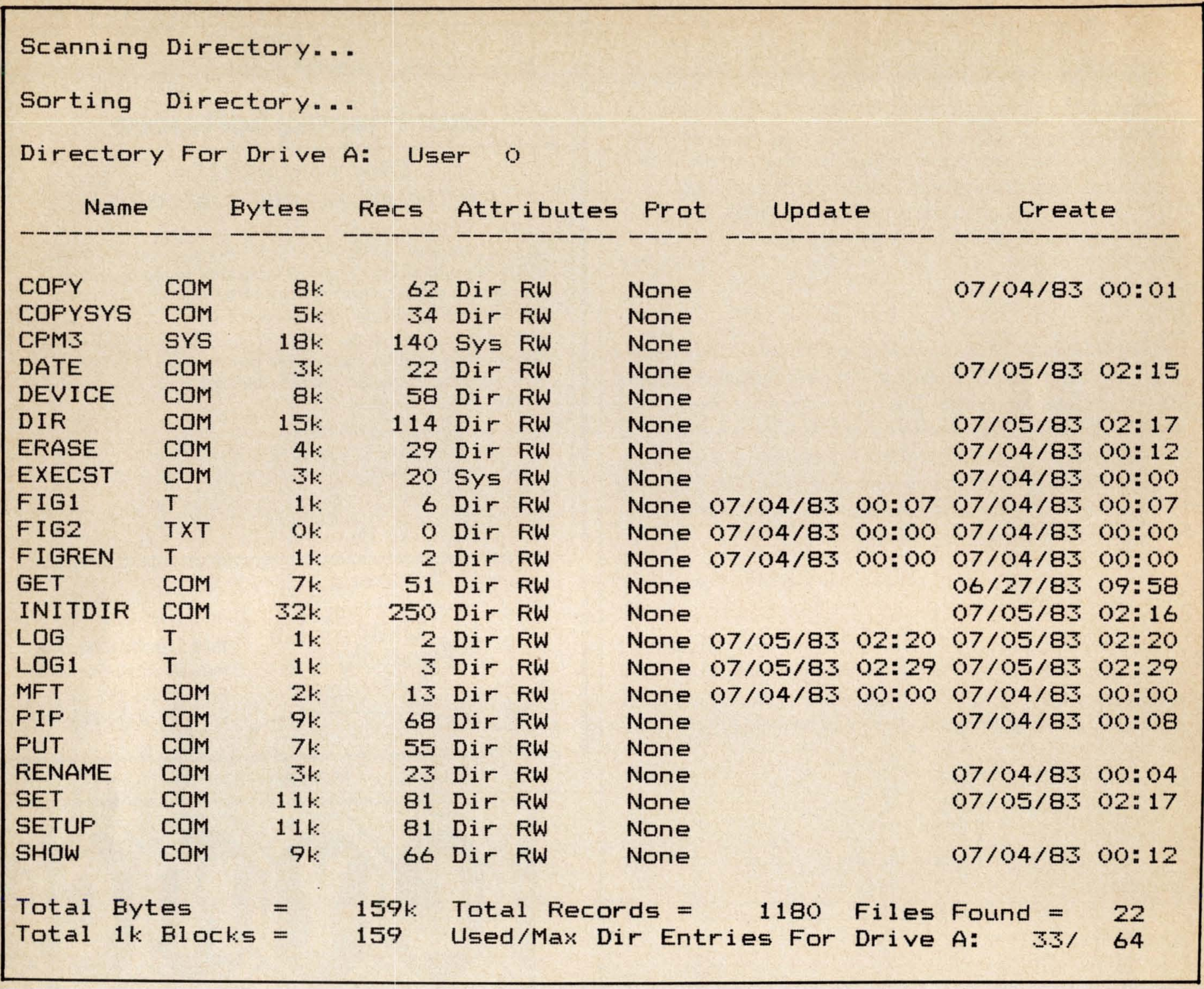
International Software Alliance (805) 966-3077
1835 Mission Ridge Santa Barbara, CA 93103

• DEALER AND OEM INQUIRIES INVITED •

FORMATS: 8" Standard; IBM PC DOS; TRS-80 Mod II with CP/M; VectorGraphic; Cromemco; North Star; Osborne; Xerox 820 (8" or 5"); SuperBrain; NLS KayPro; TeleVideo; TRS-80 Model I or III; Zenith/Heathkit; Victor 9000; Sanyo; Altos; Fujitsu; Otrona.

CP/M is a trademark of DIGITAL RESEARCH. LYNC is a trademark of MIDNIGHT SOFTWARE.

Circle number 11 on the reader service card.



```
Scanning Directory...

Sorting  Directory...

Directory For Drive A:  User  0

    Name      Bytes  Recs  Attributes Prot    Update         Create
------------ ------ ------ ---------- ----- -------------- --------------

COPY     COM     8k     62 Dir RW     None                07/04/83 00:01
COPYSYS  COM     5k     34 Dir RW     None
CPM3     SYS    18k    140 Sys RW     None
DATE     COM     3k     22 Dir RW     None                07/05/83 02:15
DEVICE   COM     8k     58 Dir RW     None
DIR      COM    15k    114 Dir RW     None                07/05/83 02:17
ERASE    COM     4k     29 Dir RW     None                07/04/83 00:12
EXECST   COM     3k     20 Sys RW     None                07/04/83 00:00
FIG1     T       1k      6 Dir RW     None 07/04/83 00:07 07/04/83 00:07
FIG2     TXT     0k      0 Dir RW     None 07/04/83 00:00 07/04/83 00:00
FIGREN   T       1k      2 Dir RW     None 07/04/83 00:00 07/04/83 00:00
GET      COM     7k     51 Dir RW     None                06/27/83 09:58
INITDIR  COM    32k    250 Dir RW     None                07/05/83 02:16
LOG      T       1k      2 Dir RW     None 07/05/83 02:20 07/05/83 02:20
LOG1     T       1k      3 Dir RW     None 07/05/83 02:29 07/05/83 02:29
MFT      COM     2k     13 Dir RW     None 07/04/83 00:00 07/04/83 00:00
PIP      COM     9k     68 Dir RW     None                07/04/83 00:08
PUT      COM     7k     55 Dir RW     None
RENAME   COM     3k     23 Dir RW     None                07/04/83 00:04
SET      COM    11k     81 Dir RW     None                07/05/83 02:17
SETUP    COM    11k     81 Dir RW     None
SHOW     COM     9k     66 Dir RW     None                07/04/83 00:12

Total Bytes     =    159k  Total Records =     1180  Files Found =   22
Total 1k Blocks =    159   Used/Max Dir Entries For Drive A:    33/  64
```

Figure 2. *Full DIR display showing file size in bytes and records, the file attributes, the protection mode, and the time and date stamps. If you are not using time/date stamps or protection modes, the display looks like figure 1.*

includes information on these items, which cannot fit in two columns, so one full-screen column is used.

Other options we find useful are the *[FF]* and *[LENGTH]* options for listing the directory on our printer. We "wake up" the printer with a Control-*P*, and use the *[FF]* option to send an initial form feed to the printer to start a new page. The *[LENGTH]* option lets us specify the length of the page; the system prints as much of the DIR display as it can fit on one page as defined by *[LENGTH]*, then it moves to the next page, prints a new heading, and continues printing the DIR display.

DIR Information

The full display shows a lot of information, but it doesn't answer our first question: how much disk space is left? By this we mean, how much space is still available for new files? The SHOW command answers this question with a simple count of the number of 1K bytes.

"1K" stands for 1024, so "1K bytes" is 1024 bytes or "one kilobyte." File and disk space is measured in terms of kilobytes, each of which can hold 1024 characters of information (including spaces, punctuation marks, digits, special codes, etc.).

DIR shows the number of kilobytes *per file*. The file size is a necessary bit of information for transferring files — will the file fit on the other disk, how long will it take to transmit the file, is there enough room on disk to add to this file, etc.

The Bytes column shows the file size in 1K bytes, or kilobytes. The Recs column shows the file size in *records*, which are logical 128-byte sectors within the file. Although a file's actual records may occupy less than a full kilobyte, at least one kilobyte (and possibly more) is allocated for the records.

If records are added and the file grows beyond one kilobyte, another kilobyte (or more) is allocated. The amount of space allocated depends on the specific version of your CP/M Plus system (systems are modified for different disk formats and computers).

The Attributes, Prot, Update and Access and Create headings are described later with the SET command options, under "File Attributes" and "Time and Date Recording."

DIR Utility Options

There are so many DIR options that we can only summarize them. In many cases you will want to use the *[FULL]* option, or the *[DRIVE=ALL,USER=ALL]* option. The following options are shown without the usual square brackets.

ATT
Display each file's attributes (see "File Attributes").

DATE
Display the date and time stamps for each file. This option only works if the date and time stamps are active — see "Time and Date Recording."

DIR
Display only filenames that have the DIR attribute — most files have this attribute unless you have explicitly set the SYS attribute — see "File Attributes."

DRIVE=ALL
Display filenames on all on-line drives. This command may cause an error or "system hang" if a drive considered "on-line" by the system does not have a disk in it.

DRIVE=(A,B,C,...,P)
Display filenames on the drives specified. This command may cause an error or "system hang" if you specify a drive that does not have a disk in it.

DRIVE=d
Display filenames on the drive specified by *d*. This command may cause an error or "system hang" if you specify a drive that does not have a disk in it.

EXCLUDE
Display filenames that do *not* match the filenames or filename matches specified in the command line. You can use the asterisk (*) to match any number of characters, and question-mark (*?*) to match any single character, when forming a filename match.

FF
Send an initial Form Feed code to the printer (if the printer has been activated by ↑*P*).

FULL
Show the name, size, number of 128-byte records, and attributes of the files. If there is a directory label on the drive, *DIR* with *FULL* shows the password protection mode and the time/date stamps. If there is no directory label, it displays two filename entries on a line, omitting the password and time/date stamp columns. The display is alphabetically sorted.

LENGTH=n
Display *n* lines or print *n* lines before inserting a heading. Use a number between 5 and 65536 for *n*; otherwise, the default is one screen (23 lines).

MESSAGE
Display the names of drives and user areas searched while looking for the files. You can set your own disk search order with the SETDEF utility.

NOSORT
Display filenames in the order found on the disk, not sorted in alphabetical order. DIR with no options produces this type of display. DIR with options but without *NOSORT* in the option list produces an alphabetical display. Specify *NOSORT* in the option list to produce a non-alphabetical display with options.

RO
Display only the filenames that have the RO (read-only) attribute.

RW
Display only the filenames that have the RW (read-write) attribute.

SIZE
Display the filename and size in kilobytes (1024 bytes).

SYS
Display only the filenames that have the SYS attribute.

USER=ALL
Display all filenames (or filename match) in all user areas for the default or specified drive.

USER=n
Display the filenames in user area *n*.

USER=(0,1,...,15)
Display filenames in the user areas specified.

DIR Utility Examples

The following are examples from a system that has a disk label, file attributes, passwords, user areas and everything else. We do not recommend you use all of these features in a small personal computer with floppy disks as the only storage medium. Some of these features were meant for hard disks that have large amounts of storage space, where labels, passwords, attributes and user areas provide convenience and security from other users.

```
A>DIR B: [RO,SYS] ↩
```

— displays all filenames in user area 0 on drive B with RO (read-only) and SYS (system) attributes.

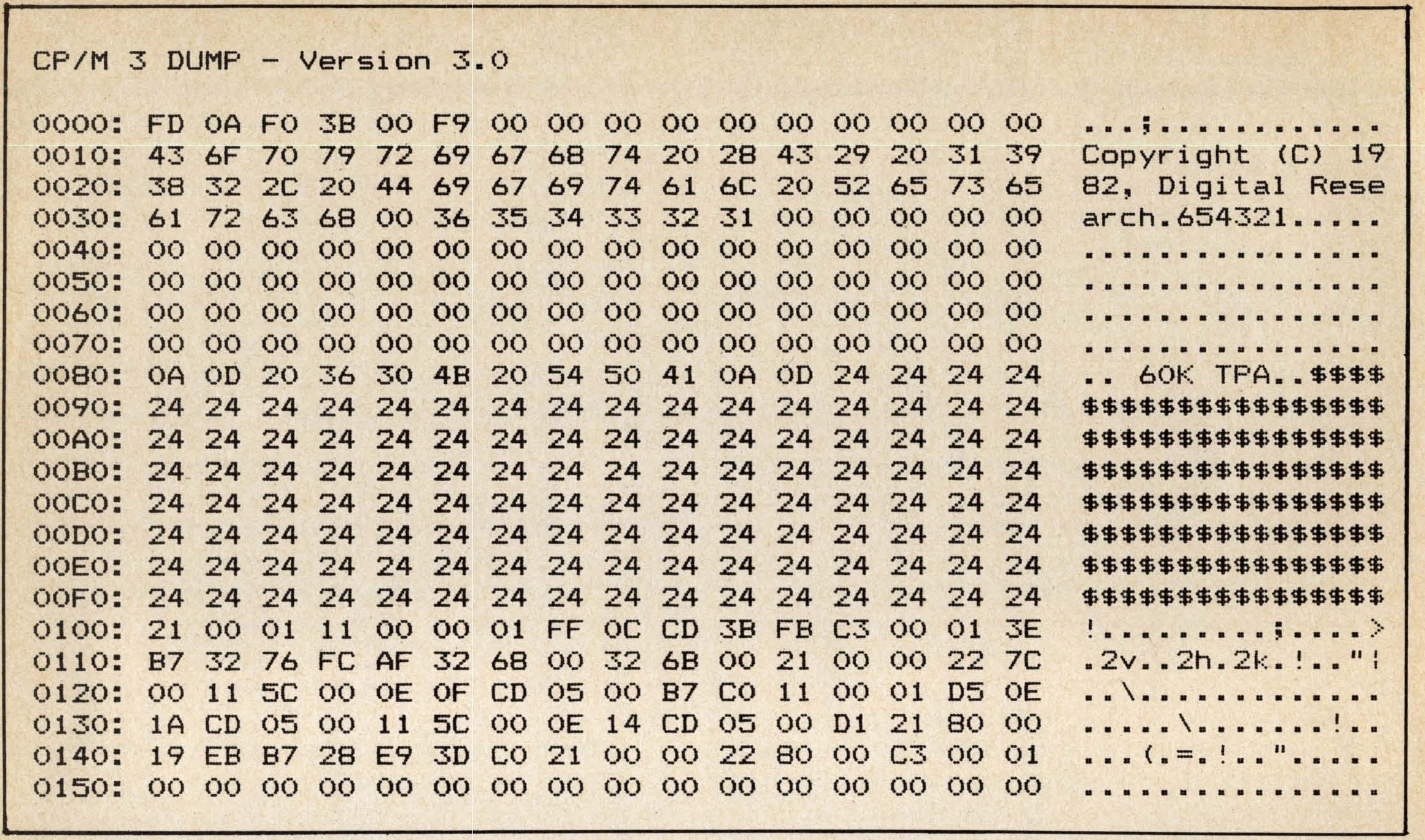
```
3A>DIR [USER=ALL,DRIVE=ALL] ↩
```

— displays all files in all user areas (0-15) on all on-line drives. The current drive is A, and the current user area is 3.

```
6B>DIR [EXCLUDE] *.DAT↩
```

— displays all the files in user area 6 on drive B that do *not* have the ".DAT" extension.

```
3B>DIR *.DBF *.NDX *.CMD [SIZE] ↩
```



```
CP/M 3 DUMP - Version 3.0

0000: FD 0A F0 3B 00 F9 00 00 00 00 00 00 00 00 00 00  ...;............
0010: 43 6F 70 79 72 69 67 68 74 20 28 43 29 20 31 39  Copyright (C) 19
0020: 38 32 2C 20 44 69 67 69 74 61 6C 20 52 65 73 65  82, Digital Rese
0030: 61 72 63 68 00 36 35 34 33 32 31 00 00 00 00 00  arch.654321.....
0040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0080: 0A 0D 20 36 30 4B 20 54 50 41 0A 0D 24 24 24 24  .. 60K TPA..$$$$
0090: 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24  $$$$$$$$$$$$$$$$
00A0: 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24  $$$$$$$$$$$$$$$$
00B0: 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24  $$$$$$$$$$$$$$$$
00C0: 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24  $$$$$$$$$$$$$$$$
00D0: 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24  $$$$$$$$$$$$$$$$
00E0: 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24  $$$$$$$$$$$$$$$$
00F0: 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24  $$$$$$$$$$$$$$$$
0100: 21 00 01 11 00 00 01 FF 0C CD 3B FB C3 00 01 3E  !.........;....>
0110: B7 32 76 FC AF 32 68 00 32 6B 00 21 00 00 22 7C  .2v..2h.2k.!.."|
0120: 00 11 5C 00 0E 0F CD 05 00 B7 C0 11 00 01 D5 0E  ..\.............
0130: 1A CD 05 00 11 5C 00 0E 14 CD 05 00 D1 21 80 00  .....\.......!..
0140: 19 EB B7 28 E9 3D C0 21 00 00 22 80 00 C3 00 01  ...(.=.!..".....
0150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Figure 3. *A "dump" of a file's actual contents. The DUMP command shows you what non-text files actually look like, in hexadecimal numbers with an extra column containing any readable ASCII text it can find.*

— displays the names and sizes of all files with the ".DBF", ".NDX" and ".CMD" extensions in user area 3 on drive B.

```
A>DIR MYFILE.TXT [DRIVE=ALL,USER=ALL] ↩
```

— displays the filename MYFILE.TXT if it exists on any active drive in any user number.

```
A>DIR B: [SIZE,RW] ↩
```

— displays the name of each file found on drive B with the RW (read-write) attribute, with the file size in 1K bytes (kilobytes).

Displaying File Contents: TYPE and DUMP

CP/M Plus is smarter than CP/M 2.2 because it assumes you are using a video terminal and pauses scrolling after filling the screen, giving you a chance to read those vital messages before they vanish. You can turn this paging off using the *[NOPAGE]* option, and turn it back on again with the *[PAGE]* option.

With paging, the system pauses after a screen is filled with output, and continues when you press any key on the keyboard. If you want to change the screen's "page" length, you must first use the SET utility before you use the TYPE command. (You may need to change the page length for "echoing" files to the printer.)

The CP/M Plus version of the TYPE command lets you type the command without expecting you to also specify a filename. The TYPE program will prompt you for the filename if you don't specify one:

```
A>TYPE↩
Filename:SAMPLE.TXT↩
```

TYPE only works with files that contain text (in the ASCII text format used by most computers). To display the contents of a file that contains something other than ASCII text, use the DUMP command. DUMP displays the contents in hexadecimal numbers, with a column on the right showing any readable ASCII text in the file. Try DUMP on the file CPM3.SYS on your system disk:

```
A>DUMP CPM3.SYS↩
```

DUMP can transfer the contents of any file, in hexadecimal form, to the console, to the printer if you type a Control-*P* first, to another device if you first change devices with the DEVICE command, or to a file if you first use the PUT utility.

DUMP is useful for identifying strange non-text files, because if the file is a program with a copyright notice, the notice should be readable and should appear in the right column.

If you intend to change bytes in a non-text file with a "debugger" or disk utility program, you would want a "dump" of the file on the printer to serve as a map to the data.

Most non-programmer users have no need for DUMP.

Copying Files

The PIP utility can copy files from one disk to another, or copy the contents of a file to another file on the same disk, or copy the contents of a file to a device like the printer or modem. PIP can also join two or more files and copy the result to a third file.

CP/M Plus added two other features to PIP: the *[A]* option to selectively copy only files that have changed, and an advanced *[Gn]* option that lets you copy *to* a different user area as well as copy *from* one.

All PIP expressions contain a *source* for the copy and a *destination* for the copy, in this fashion:

```
PIP destination = source ↩
```

The following are examples of file copying:

1. Copy the *original* file to the *copy* file on the same disk:

```
A)PIP COPY.TXT=ORIGINAL.TXT↩
```

2. Copy SOURCE.BAS on the disk in drive A to DESTINY.BAS on the disk in drive B, creating a new DESTINY.BAS or overwriting the old one:

```
A)PIP B:DESTINY.BAS=A:SOURCE.BAS↩
```

3. Copy ORIGINAL.TXT in drive A to drive B, using the same filename for the copy in drive B:

```
A)PIP B:=A:ORIGINAL.TXT↩
```

4. Copy ORIGINAL.TXT in drive A to drive B, using the *[Z]* option to "zero" the parity bit of each byte (i.e., turn a WordStar document file into a non-document file):

```
A)PIP B:=A:ORIGINAL.TXT [Z] ↩
```

Copying An Entire Disk or User Area

This example copies the entire disk in drive A to drive B (user areas not used):

```
A)PIP B:=A:*.*↩
```

To copy an entire disk where files are grouped into user areas, use the *[Gn]* option, substituting for *n* the number of the user area containing the source files. Here's the PIP command for copying from user area number 2 on drive A to the current user area (user 0) of drive B.

```
B)PIP B:=A:*.* [G2] ↩
```

To copy *to* a different user area, place the *[Gn]* option on the left side of the equal sign:

```
B)PIP B: [G3] =A:*.* [G2] ↩
```

This example copies all of the files from user area 2 on drive A to user area 3 on drive B.

Copy With Confirmation

CP/M Plus provides a new option that lets you specify all files on a disk and *confirm* which files to copy on a file-by-file basis. For example, this command causes PIP to ask for a confirmation before copying each file with the ".TXT" extension:

```
A)PIP B:=A:*.TXT [C] ↩
```

With each confirmation message, type *Y* to copy the file or *N* to skip the file and go to the next one.

Keep Magazines, Catalogs, Manuals, Journals

NEAT & ORGANIZED

End Clutter In Home, Office, Workshop With

SHELF FILES

Eliminate the mess of loose magazines, catalogs and newspapers. Find what you want when you want it by using these handy shelf or desk top files. Attractive brown front panel. Adhesive ID labels included.

6 POPULAR SIZES AVAILABLE

From digest size to newspaper size! Popular letter and magazine size (9" x 11½", 3¾" box width) – 8 for $13.95. 25 for $31.95 prepaid. Sturdy, heavy-duty fiberboard will last for years! Charge it! Call or mail coupon today.

Professional Aids Co., Suite CP73
1678 S. Wolf Road, Wheeling, IL 60090 • 312/459-6828

Please rush postpaid Fiberboard Desk and Shelf Files as checked below:

How Many	Qty.	Size	Price	Total Price
	8	Letter Size	$13.95	
	25	Letter Size	$31.95	

☐ Check for Free Catalog.
I enclose $____ full payment. Ship postpaid.

Illinois Residents add 6% Sales Tax, please. Other sizes from digest to newspaper available.

☐ Charge it to ☐ VISA ☐ MasterCard TOTAL: ____

Card No. ____ Exp. Date ____

Name ____

Address ____

City, State, Zip ____

Circle number 14 on the reader service card.

Joining Files

You can join (*concatenate*) one or more files to the end of a source file and copy the result to a destination file. The following example adds the contents of FILE3 to the end of FILE2, and FILE2 to the end of FILE1, and stores the joined contents in RESULT without disturbing FILE1, FILE2 or FILE3:

```
A)PIP RESULT=FILE1,FILE2,FILE3↩
```

Archiving Files

Applications involving large hard disk systems will find this feature useful. Each file has an *archive attribute* which acts like a flag signaling that the file has changed — has had something new written to it.

PIP when used with the *[A]* option copies only the files that have changed since the last PIP operation with the *[A]* option. If you adopt the habit of archiving files on a regular basis, this feature saves you time by copying only the files that have changed.

To apply the "archive" option to all files on the disk, type the *.* filename match as the source:

```
A)B:=A:*.*[A] ↩
```

PIP with the *[A]* option automatically resets the archive attribute flag for each file it copies, marking it as a file that has been copied. If the flag is set by a new change, or by use of the SET utility (described later) which can alter any file attribute, PIP with the *[A]* option will copy the file again.

Renaming Files

The most useful change to occur in CP/M is the expansion of the RENAME utility. The REN built-in command hasn't changed — it still lets you rename a single file in the same manner:

```
A)REN NEWNAME.TXT=OLDNAME.TXT↩
```

The RENAME utility lets you change the name of a group of files at once, as well as one file at a time. Also, if the new name is the same as an existing file, you are given a choice of erasing the existing file or not using the new name for the file you are trying to rename.

Most people find the old REN command hard to remember — which filename is first, the new one or the old one? The new RENAME helps by prompting you for the new name and the old name, in that order. To use the RENAME utility you must have a copy of RENAME.COM available, on the current disk in user area zero (where it can be used from other user areas), or on another disk in the search order set by SETDEF.

In the following example, we use RENAME to find all files that end with the ".TXT" extension and rename them to end with the ".615" extension; RENAME displays the files as it renames them:

```
A)RENAME↩
Enter New Name: *.615↩
Enter Old Name: *.TXT↩

FIG1          .615=FIG1          .TXT
FIG2          .615=FIG2          .TXT
LOG           .615=LOG           .TXT
```

In this example you can see that RENAME finds each file with the ".TXT" extension and simply renames the extension part of the filename, leaving the primary name as it was.

You can abbreviate *RENAME* to *REN*, and the RENAME utility is executed if it is found on disk (otherwise, the built-in REN command is executed). The following example shows how you can rename part of the primary name, as long as the new names are unique to each other:

```
A)REN↩
Enter New Name: PIC?.615↩
Enter Old Name: FIG?.615↩

PIC1          .615=FIG1          .615
PIC2          .615=FIG2          .615
```

RENAME does not make a copy of a file, it simply renames the directory entry for the file. You can rename files on another disk by specifying the drive in the old filename or new filename (e.g., *B:FIG1.TXT*); however, since you can't use rename to copy a file to another disk, you can't use a different drive for each name.

Erasing Files

It used to be *too easy* to erase files in a CP/M system. Once erased, the files are gone forever unless you have an "un-erase" program or a disk utility that fixes disk problems and recovers erased files.

CP/M Plus offers a new ERASE utility to complement the simple ERA built-in command. You can abbreviate *ERASE* to *ERA*, and the ERASE utility is executed if it is found on disk (otherwise, the built-in ERA command is executed). To use the ERASE utility you must have a copy of ERASE.COM available, on the current disk in user area zero (where it can be used from other user areas), or on another disk in the search order set by SETDEF.

ERASE can still erase files as easily as ERA:

```
A)ERA FILE.TXT↩
```

In addition, ERASE.COM has some safeguards to prevent you from accidentally wiping out something you really want to keep. If you provide a filename match to select sever-

```
Label for drive B:

Directory         Passwds  Stamp    Stamp    Stamp
Label             Reqd     Create   Access   Update
---------------   -------  -------  -------  -------
B:LABEL     .       on       on       off      on
```

Figure 4. *Display of the disk label using the SHOW command with the [LABEL] option.*

al files for erasing, ERASE asks you to confirm before erasing any files:

```
A)ERA *.TXT↩
ERASE *.TXT (Y/N)?Y↩
```

You can type *N* to stop the erase operation.

ERASE also gives you the *[CONFIRM]* option. When you use this option, ERASE asks for a confirmation before erasing *each* file. This option is especially useful for erasing some but not all of the ".TXT" files — typing *Y* to confirm the ones to erase, and *N* to leave other ones intact:

```
A)ERA *.TXT [CONFIRM] ↩
ERASE FIG1.TXT (Y/N)?Y↩
ERASE FIG2.TXT (Y/N)?Y↩
ERASE GOOD1.TXT (Y/N)?N↩      — Save this one.
ERASE GARBAGE.TXT (Y/N)?Y↩
ERASE INFOAGE.TXT (Y/N)?N↩— Save this one also.
```

Both versions of ERASE remove a file by substituting the hexadecimal number E5 (E5H) for the first byte of the directory entry, usually used for the user area number. An E5H in this byte indicates an erased file, and the disk sectors are from that point on "up for grabs" for new files or the expansion of existing files.

If a file is erased and no other disk "writes" (changes, updates or deletions) have occurred, you can recover the erased file by changing the E5H byte to zero. There are several disk editing utilities (such as the public domain DU.COM, described in issue no. 4 of *User's Guide.*) available from various sources.

SHOW Disk Status

The SHOW command displays information on the overall status of disks and drives, and lets you see a normally invisible entry in the disk directory that acts as a disk label (CP/M 2.2 users will recognize the other half of STAT in the SHOW program).

SHOW by itself displays the amount of free disk space (space not occupied by files) for all drives currently active in the system. *SHOW* also tells you the "access mode" of the disk — RW for read-write (changes are allowed), or RO for read-only (writes or changes are not allowed).

The *[USER]* option displays the current user area number and the number of files in each user area:

```
A)SHOW [USER] ↩
A:Active User          : 0
A:Active Files         : 0      1     2
A:# of files           : 24     1     1
A:Number of time/date directory entries:  16
A:Number of free directory entries:  19
```

This example shows the number of files in each user area. The user area numbers displayed are ones with "active files." The "active user" is the area you are currently using. The example shows an Osborne Executive floppy disk with user areas 0, 1 and 2. There are 24 files in user 0, and one file each in user areas 1 and 2. There are 16 directory entries with time/date stamps, and 19 free directory entries (this number determines the number of additional filenames you can fit in the directory).

When using a hard disk, these numbers will be more useful. You may have several user areas with forty or fifty files in each user area. SHOW helps you find the more populated user areas easily. It helps to put the SYS attribute on the file SHOW.COM to make it available from any user area.

SHOW with the *[LABEL]* option shows the display in figure 4. The Directory Label is the label you can assign to your disk with the SET command. You can also turn on password protection and the time/date stamps with the SET command.

SHOW with the *[DRIVE]* option displays the disk drive characteristics (CP/M 2.2 users will recognize this display from the STAT DSK: command):

```
A)SHOW [DRIVE] ↩
       A:  Drive Characteristics
   1,096:  128 Byte Record Capacity
     137:  Kilobyte Drive Capacity
      64:  32 Byte Directory Entries
      64:  Checked Directory Entries
     128:  Records / Directory Entry
       8:  Records / Block
      40:  Sectors / Track
       3:  Reserved Tracks
   1,024:  Bytes / Physical Record
```

Here's a summary of the SHOW utility options (without brackets):

SPACE

Display the disk access mode and space. *SHOW* by itself does this, and this option is provided to include in a list of other options, in order to include this display with the other option displays.

LABEL

Display the disk label and the time/date stamp modes.

USERS

Display the current user area number, and all user areas with active files on the drive, with corresponding number of files assigned to them.

DIR

Display the number of free directory entries.

DRIVE

Display detailed drive characteristics.

Here are examples:

```
A⟩SHOW B: [LABEL,SPACE] ↵
```

— displays label information and disk space for drive B.

Disk and File Setup Utilities

Systems with a hard disk typically have four or more logical disk drives: three for hard disk (perhaps A, B, & C) and one or more for one or more floppy disk drives or removable cartridges. Hard disk systems have capacities that require some kind of organization, which is why three drives are assigned to one physical disk: to separate the files in some logical manner.

This separation may not be enough. Files within each drive can be further separated into *user areas*. You copy files from one user area to another — with such expansive disk space, there's no need to delete the original copies. This leads to problems of a different kind: which copy of the file is the most up-to-date?

Hard disk systems are also more expensive, and perhaps less personal. You may want protection for your files. Passwords are the typical approach. CP/M Plus lets you assign a label and a password to each drive, a password to each file, and three levels of protection for each file.

Date and Time

The system will keep track of the time and date for as long as it remains on or until you perform a "cold start" or reset. Therefore, if you use time and date recording on a disk, you *must always* type an accurate time and date when you turn on the system or re-start it and use that disk.

Time and date information can be quite useful on disks in large systems that are on most of the day. They can be also be useful on archive disks that are seldom used except under controlled copying operations with accurate time and date. However, if you don't supply the correct time and date when you start or re-start the system, each file's time and date will probably change (when activated) to December 1982, or whatever date the utility was created.

To "turn on" time/date recording, you set the time and date, then INITDIR ("initialize") the disk directory, and finally SET the recording mode.

Your **first step** is to set the system clock with the *DATE* utility:

```
A⟩DATE 07/16/83 22:05:00↵
```

This example sets the date to July 16, 1983, and the time to 10:05 PM (22nd hour of the day). If you typed *10:05:00* for 10 PM, the system would assume it was 10 AM and it would not automatically change the date after midnight.

Date can display the time and date continuously. If you use the *SET* option, DATE prompts you for the time and date (useful if using DATE in a "submit" file). Here are examples of DATE with options:

```
A⟩DATE↵
```

— displays the current date and time.

```
A⟩DATE C↵
```

— displays the date and time continuously.

```
A⟩DATE SET↵
```

— displays prompts for date and time entries.

Initializing the Directory

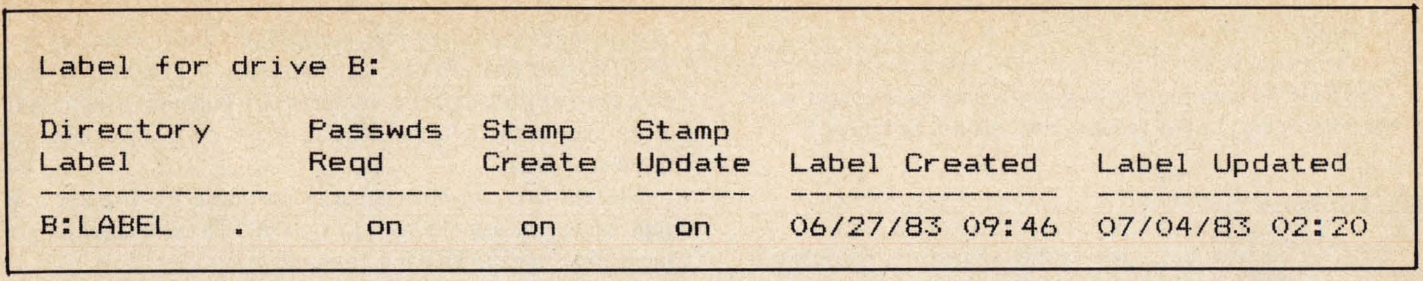
The **second step** is to create the directory space on your disk for the time/date information. The INITDIR utility "initializes" a disk directory to either make room for time and date information, or clear out the time and date information.

When you use INITDIR on a new disk (most typical case), it creates the space for the time and date. Usually INITDIR is used to initialize the directory of a disk in drive B or a drive other than the current drive, although this is not required.

```
A⟩INITDIR B:↵
INITDIR WILL ACTIVATE TIME STAMPS FOR THE SPECI-
FIED DRIVE.
Do you want to re-format the directory on drive: B (Y/N)?
```

Type a *Y* or *y* to re-format the directory and activate "time stamps." However, another message may appear if the disk you are initializing already contains time/date information:

```
Directory already formatted.
Do you want to recover time/date directory space (Y/N)?
```



```
Label for drive B:

Directory       Passwds  Stamp    Stamp
Label           Reqd     Create   Update   Label Created     Label Updated
--------------  -------  -------  -------  --------------    --------------
B:LABEL    .       on       on       on    06/27/83 09:46    07/04/83 02:20
```

Figure 5. *The disk in drive B has now been SET to use the CREATE and UPDATE time/date modes, to record the time and date of each file's creation and modification.*

This message means that INITDIR was already used once on this disk to set up time/date recording. To eliminate time/date recording ("stamping") from your disk, type *Y* or *y* to recover the directory space (to add them later, you must use INITDIR first).

However, if you simply want to clear existing times and dates and allow new time and date information, answer no (*N* or *n*). INITDIR then asks:

```
Do you want the existing time stamps cleared (Y/N)?
```

Answer with *Y* or *y* to clear the existing time/date "stamps" but keep "stamping" active. Answer *N* or *n* to leave time/date "stamps" as they are.

Time and Date Recording

The *third step* is to use the SET utility to activate the time/date recording modes. You can activate *update mode* and one of two other modes: *create* or *access*.

Update mode means a new time and date is recorded with the filename whenever the file is changed or added to (any "writes" to the file). This is the most useful time/date recording mode.

There are two columns in DIR displays for these recording modes: update mode, and create or access mode.

In *access mode*, a new recording occurs whenever you display or access the file. In *create mode*, the system records the date of new file creations. To record the time and date of a file's creation, this mode must be in use before creating the file.

We use the create and update modes with most data and backup disks. With the time and date set properly, we format a new data disk and backup data disk, run INITDIR, and type the following SET command for each disk:

```
A)SET B: [CREATE=ON,UPDATE=ON] ↩
```

This form of the SET command displays a message like the one in figure 5, showing the name of the disk (under the heading Directory Label) and the various *modes* — protect-unprotect mode (Passwrds Reqd on or off), time/date create mode (Stamp Create), time/date access mode (Stamp Access), and time/date update mode (Stamp Update).

With these modes on, we use the data disk in an application such as SuperCalc, where spreadsheet data files change rapidly over a period and the time and date information is a clue to the most recent versions of files on the backup disks. In this application, the access mode is not meaningful.

Access mode is meaningful on system disks containing program files and utilities. Each use of a program or utility is recorded with a new time and date.

With recording modes active, the *DIR [FULL]* display has new information (as shown back in figure 2.) You will notice that the creation time and date as well as the update time and date will change for files used with text editing programs like ED and WordStar.

This is because many text editing and other programs do not change the file — they create a temporary new file, and make all corrections and additions in this temporary file. When you "save the file," the program renames the original file to have a ".BAK" extension (for "backup"), and renames the new temporary file to have the name of the original file. Thus, the same filename receives a new creation time and date.

Overall, time and date information is useful in hard disk systems where users share files, or in certain personal applications that generate a lot of data files on a lot of disks.

The Drive Attribute

Occasionally you may want to slap a temporary curfew on a disk prohibiting any changes or deletions. Although floppy disks can be semi-permanently protected from writes or deletions with the write-protect tab (remove the tab on eight-inch disks, or apply the tab on 5.25-inch disks), you may not want to harm the disk by attaching and then removing a write-protect tab for a temporary situation. Also, hard disks have no such "hard" protection as a physical tab.

CP/M offers a "soft" write-protection feature called the *disk attribute* or *disk access mode* (depending on which manual you refer to). For example:

```
A)SET B: RO↩
```

This command sets the disk drive B: attribute to read-only (*RO*). Drives are usually read-write (*RW*), which means that "writes" (changes or deletions) can occur. If a drive is read-only, you cannot rename, erase or change files, nor can you copy to the disk until you type a Control-C (↑*C*) to reset all drives to read-write.

Another way to set a single disk drive back to read-write access is the following command:

```
A>SET B: RW↩
```

The SHOW command tells you whether each active drive is RO (read-only) or RW (read-write, the normal mode).

File Attributes

CP/M Plus provides a new archiving feature that lets you keep an *archive attribute* for each file that acts as a flag to tell whether or not the file needs to be archived. The attribute flag changes if a change occurs to the file, signaling that the file now needs to be archived since it contains updated information.

Files can also be protected and hidden from DIR displays using *file attributes*. (This protection does not include the password protections available in the banked version of CP/M Plus, described next.)

For example, you can set a file to be "read-only" (no changes, deletions or renaming allowed) with the following command:

```
A>SET B:WATERGAT.TAP [RO] ↩
```

WATERGAT.TAP on drive B is now protected from an accidental ERA command or any changes to the file. If you were to specify WATERGAT.TAP as the destination of a copy in a PIP command, you would get a message asking if you want to delete the file. Type *N* to keep WATERGAT.TAP intact, or *Y* to replace the contents of WATERGAT.TAP with new information. PIP also has the *[W]* option that lets PIP overwrite a read-only file without asking this question.

Read-only protection lasts until you press Control-*C* or you change it with the following command:

```
A>SET B:WATERGAT.TAP [RW] ↩
```

This command sets the file access attribute to *RW* (read-write), the normal mode for most files.

The archive attribute is either *ON* or *OFF*. *ON* tells the PIP command (when you use PIP's *[A]* option) not to copy the file because the file has not changed:

```
A>SET B:WATERGAT.TAP [ARCHIVE=ON] ↩
```

This feature is useful for making sure that a file or group of files is not copied when you are using PIP with its *[A]* option.

OFF marks the file ready for backup, so that the PIP command with the *[A]* option copies the file:

```
A>SET B:WATERGAT.TAP [ARCHIVE=OFF] ↩
```

You would use this feature to make sure a file is included in the list of files to be copied under the PIP *[A]* option, whether or not the file has changed.

PIP, when you use the *[A]* option with it, automatically sets the archive attribute "on" after copying the file.

The other two attributes are *DIR* (directory display) and *SYS* (system file). When you designate a program in user area zero as a system file, the program becomes available to other user areas without requiring you to make copies for those user areas.

System files are not visible in normal DIR displays (except when you use the *[FULL]* option). To see a directory of these files, use the DIRS command.

The *DIR* attribute is the "flip-side" of the *SYS* attribute — use it to turn a system file into a regular file that shows up in DIR displays.

Here's a summary of file attributes:

DIR — changes a system file (*SYS*) to a directory file.

SYS — changes a directory file (*DIR*) to a system file.

RO — changes a read-write enabled file (*RW*) to a read-only file.

RW — changes a read-only file (*RO*) to a read and write enabled (changeable) file.

ARCHIVE=OFF — sets the archive attribute to "off," marking the file ready for backup using the PIP command with the *[A]* option. PIP sets the archive attribute "on" after copying the file.

ARCHIVE=ON — sets the archive attribute to "on," so the file will not be copied in your next PIP operation using the *[A]* option.

F1=ON or *OFF* — turns on or off a custom file attribute. You can have attributes *F1* to *F4*.

Disk Passwords (Banked Version Only)

Before setting file passwords, you should set a *disk password*. You can set file passwords at any time, but they can be changed by someone else unless you hide the SET utility, or use a *disk password* to protect the SET utility from use.

Basically, the file's password can prevent someone from reading your file, or just prevent it from being changed or erased. The disk's password protects use of the SET utility, which can turn off passwords or change them.

Think of the disk password as the ultimate password to use the SET utility. Since SET is the only utility that sets or changes passwords, the disk password is the key to all passwords. If you *don't* use a disk password and you share disks with other users, you may find yourself locked out of the disk because someone else created a disk password.

The "catch-22" for using disk passwords is this: if you forget your disk password, you cannot change it or erase it using the SET utility. The disk becomes a fortress, and since the password is encrypted, you can't use a conventional disk editing program to find the key.

To create a disk password, you should start by creating a disk name. If you do not assign a disk label before assigning a disk password, the system automatically uses the label Label.

Here's how to create a disk label (we use Budgets) for the disk in drive B (figure 6 shows the display after this command executes):

```
A)SET B: [NAME=Budgets] ⏎
```

Labels are just like CP/M filenames: they can have up to eight letters or digits, followed by an optional period and 3 letters or digits. You cannot use blanks between letters or digits.

Choose a label that aptly describes the disk's application. To set the disk password, use the SET command with the *[PASSWORD]* option:

```
A)SET B: [PASSWORD=THX1138] ⏎
```

After typing this command, the password *THX1138* is set as the disk password, which protects use of the SET utility. In fact, if you try to use SET to change it, you must type the password to use SET. Write down the disk password and keep it in case you forget it.

Passwords can have up to eight letters or digits. You cannot use blanks between letters or digits.

To *erase* the disk password from the disk in drive B, use this version of the SET command:

```
A)SET B: [PASSWORD= ⏎
```

File Passwords (Banked Version Only)

You can assign unique passwords to each file, or the same password to a set of files. You can also pick the type of protection: passwords required for erasing a file, changing a file, or even reading a file. Each *protection mode* is a type of file access for which a password is required.

You can also suspend the use of file passwords, or change all passwords to a default password (which can stay in effect during system operation), if you have access to the SET utility and the disk password. With this feature you could password- protect all data files on a disk, and make them available to someone else during a system session without revealing the original passwords.

Step one in this process is to activate file protection by turning the "disk protect mode" on:

```
A)SET B: [PROTECT=ON] ⏎
```

AS SOON AS I'M PROGRAMMED FOR SECURITY I'LL LOCK THIS GUY OUT
©L Shoemaker '83

This command turns on file password protection for the entire disk in drive B. You must turn password protection on before assigning passwords to files.

Always write down your passwords — if you forget a file's password, you cannot use the file unless you turn off password protection for the entire disk with the SET utility (for which you need the disk password). If you forget the disk password, you cannot turn off password protection for the disk.

To assign a password to a file, use the SET command with a filename or a match for several filenames (using the * and ? symbols):

```
A)SET B:BUDGEDIT.CAL  [PASSWORD=SPLURGE] ⏎
```

You could assign the password *SPLURGE* to all ".CAL" files with this command (you can also abbreviate *PASSWORD* to *PASS*):

```
A)SET B:*.CAL  [PASS=SPLURGE] ⏎
```

The password *SPLURGE* is now part of the filename. To specify the BUDGEDIT.CAL file on drive B, you would type:

```
B:BUDGEDIT.CAL;SPLURGE
```

The semicolon (;) separates the password from the filename.

SET automatically sets the type of protection to *read only*, which you can change. To change to another type of protection, use the *[PROTECT]* option with the SET command:

```
Label for drive B:

Directory      Passwds  Stamp   Stamp
Label          Reqd     Access  Update  Label Created   Label Updated
------------   -------  ------  ------  --------------  --------------
B:BUDGETS        on       on      on    06/27/83 09:46  07/19/83 05:35
```

Figure 6. *Display after using SET to create a new label for the disk in drive B.*

```
A)SET B:BUDGEDIT.CAL  [PROTECT=WRITE] ⏎
```

In fact, you could combine the password and its protection mode in one command:

```
A)SET  B:BUDGEDIT.CAL  [PASSWORD=SPLURGE
,PROTECT=WRITE] ⏎
```

READ mode (the default mode) is the most strict — it doesn't allow any access, including writing to, deleting, or even reading its contents. The other possible protection modes are:

WRITE — the password is required for writing, deleting or renaming the file. You do not need a password to read, copy or print the file.

DELETE — the password is required only for deleting or renaming the file. You do not need a password to read or modify the file.

READ — the password is required for reading, copying, printing, writing, deleting or renaming the file.

NONE — if a password exists, use this option to delete it.

The Default Password

If several files on disk use the same password, and you don't want to have to type it each time, you can set the password as the *default password* for the duration of the session until the next reset or cold start.

The default password is in memory, not on disk. When you omit the password to a password-protected file, the system tries to use the default password to access the file; if the default word is the same as the password stored for the file, you gain access to the file.

To set a default password, use the *[DEFAULT]* option with a disk drive specifier (or no specifier for the current drive):

```
A)SET B:  [DEFAULT=THX1138] ⏎
```

The password *THX1138* is set to the session default.

Turn Off Passwords

The following command turns off file protection for the entire disk and temporarily disables the file passwords:

```
A)SET B:  [PROTECT=OFF] ⏎
```

Your passwords are still stored for each file, but password checking is turned off with the above command.

To delete a password and turn off protection for a file, type this command:

```
A)SET B:BUDGEDIT.CAL  [PROTECT=NONE] ⏎
```

Another way to delete a password:

```
A)SET B:BUDGEDIT.CAL  [PASS= ⏎
```

This command turns the file password into blanks and automatically turns protection off for the file.

Here are examples of file passwords and the SET command:

```
A)SET B:*.TXT  [PASS=USERGUID,PROT=WRITE] ⏎
```

— assigns the password USERGUID to all the files with the ".TXT" extension on drive B. Each file is protected against writes to prevent unauthorized editing, changing, deleting or renaming. No password is required for reading, copying, or printing the files.

```
A)SET  [DEFAULT=COMMON] ⏎
```

— sets COMMON as the default system password while the system is up and running. The system uses *COMMON* to access files if you do not type a password for a password-protected file. If the file's regular password is also *COMMON*, you can access the file without specifying the password.

```
A)SET *.COM  [SYS,RO,PASS=KNOWIT,PRO- T=READ] ⏎
```

— sets the most protection for all the programs (".COM" files) on the disk in drive A: protection against reading or using in any way unless you know the password KNOWIT (unless the default password is set to KNOWIT). Even if you know KNOWIT, you cannot write to the file until you change its *RO* (read-only) attribute. The *SYS* attribute keeps the files hidden (not listed in a normal DIR display).

```
A)SET *.COM  [DIR, RW, PROTECT=NONE] ⏎
```

— removes the protection set in the previous example: no password protection on the file, the *RW* (read-write) attribute, and the *DIR* attribute for displaying the files in the directory.

User Areas

When you have a large amount of disk space for a given drive, you may want to separate files by placing them in "user areas" identifiable by number (user area 0, 1, etc.). The disk is not actually carved into areas; the numbers are added to the file's description in the directory, and you need to remember the user area to find the file.

For example, your text files may be in user area 3; if you are in user area 1, you won't see them until you issue the USER command to change the user area to 3.

One vast improvement over CP/M 2.2 user areas: files in user area 0 that have the SYS attribute can be executed from other user areas. You no longer have to keep a copy of PIP and other utilities in each user area. One copy of each system utility and application program can be placed in user area 0 and given the SYS attribute.

Here is an example of the USER command:

```
A>USER 1↩      — Switch to user area 1.
```

Each disk directory can be divided into distinct groups according to user area numbers. User area numbers range from 0 through 15. If you type *USER* by itself, you are asked for a user area number:

```
A>USER↩
Enter User4↩    — Switch to user area 4.
```

For a display of the number of active files in each user area, use the following command:

```
A>SHOW [USER] ↩
```

To see all files in all user areas, try this:

```
A>DIR [USER=ALL] ↩
```

To see files in user area 2, use this command:

```
A>DIR [USER=2] ↩
```

To copy files from one user area to another, use the PIP utility. PIP.COM must reside in the current user area, or in user area 0 with the SYS attribute. The following example copies a file from user area 4 to user area 2 (the current user area is 0):

```
A>PIP A: [G2] = FILE.TXT [G4] ↩
```

In this example, we specify drive *A:* as the destination even though the current drive is drive A. PIP requires a destination of a drive or a filename or both; the option *[G2]* by itself is not sufficient, even though it specifies user area 2 as the real destination.

Summary of Operating Utilities

Many utility programs are needed for everyday operations, and they should be copied to every system disk. If you are short on disk space (especially if you use floppy disks), choose the utilities you need for a specific application, and make "application-specific" system disks.

The "most used" utility is PIP, which lets you copy one or more files to other disks, make a copy of a file with another name, and merge the contents of several files into one file. PIP.COM belongs on every system disk, unless you can get by using another disk copying program, or you're using MFT on a single drive system. (See description of the PMC MicroMate in this issue for a description of MFT.)

The ERASE utility provides a quick confirmation check for each file before actually erasing it, giving you time for a second thought. You should put ERASE.COM on every system disk you use; however, if you're short on space, you can get by with the built-in ERA command (which only asks for confirmation when you try to delete *all* files on the disk).

You should also put RENAME.COM on every system disk. This enhanced version of the REN command lets you rename several files at once using the *filename match symbols*. TYPE.COM is also very useful for displaying files.

The DIR.COM utility is not necessary for simple directory displays, but with it you can display filenames in alphabetical order with the file sizes, attributes, protection mode and time/date information. The SHOW.COM utility (replacing the old STAT.COM) displays the amount of space left on a disk (DIR does not show this information, although many public domain programs, such as SD and XDIR, provide this information with the directory display).

SUBMIT (to be described in the next installment of this tutorial) and its companion XSUB.COM, are required on any system disk that has ".SUB" files containing "batches" of CP/M commands. DEVICE (also described in the next installment) can be left on a utilities disk to be used whenever it is necessary to change device characteristics.

COPYSYS is for copying the inner two tracks — the *system* tracks — of your system disk to another disk. This utility lets you turn a data disk into a system disk that can work in drive A if you include the CPM.SYS file on the disk. You will probably not need this utility if your disk copying program can make a copy of a system disk. However, PIP will not copy the system tracks of a disk, so if you have no other way, you must use COPYSYS.

Utilities that can be kept separate on special utility disks include DATE, DEVICE, INITDIR, SETDEF and SET. Although they are very useful, they occupy disk space, and if disk space is already limited (such as with most floppy disk based personal computers and small business computers), you will probably want to understand their uses before you copy them to every system disk.

HELP can also be set aside, unless you use it regularly. HELP.COM and its companion HELP.HLP take up a considerable amount of disk space.

DATE is required at system start-up to set the time and date for systems that require accurate time and date information.

INITDIR and SET are useful at any time to prepare a new disk for time and date information (INITDIR), and set various system options such as password protection, disk labels and file attributes.

SETDEF may only be needed once to set the system search order, and turn on or off the "page mode" that pauses every display after one screen is filled. These utilities are sometimes combined into a "submit" file of commands to be executed all at once with the SUBMIT utility. You need not have them on every system disk, but you may need to use a "system warm-up" disk with these utilities to prepare your system or new disks.

The programming utilities are not needed in everyday operations (unless you are a programmer, of course). These will be covered extensively in a future article.

Subscription Fulfillment With dBASE II : Part 1

An Overview of Data Entry, Editing, Retrieval, Listings and Mailing Label Printing

by Tony Bove & Cheryl Rhodes

dBASE II has the both a flexible and a powerful command language that lets you develop a data base application. This application note provides an overview of a subscription fulfillment system that is still in stages final development. Next issue's note will go into more detail.

Elsewhere in this issue we present part one of a tutorial on dBASE II — an introduction to creating, displaying and editing a data base. This article is the first of a series of application notes on dBASE II.

This note presents some of the methods we use to organize our subscription data base activities — from entering the data, performing routine editing, combining subfiles into a major file, selecting data into subfiles by postal zone, and printing mailing labels in zip code order for each postal zone.

For each of these activities we created a dBASE II *command file* using the dBASE II command language. The files we present here are simpler versions of the files we use in conjunction with an overall menu program.

We don't show the menu program yet, because we designed the smaller command files first, and you'll probably want to approach an application in the same manner — designing each individual task to run by itself. The tasks are combined later into a menu system to make it easier for any typist to run the tasks; however, the menu system is a topic for another application note.

Designing a Structure

A quick tutorial in this issue describes how to start dBASE II and define a structure for a data base. A data base structure usually reflects both the function of the data base and the personality of the data base user. Although the structure is the most important aspect of designing a data base application, you shouldn't decide on a final structure until you've experimented with routines such as label printing.

dBASE II is a trademark of Ashton-Tate (Culver City, CA).

Before deciding the final structure of a subscription fulfillment system, we had to define the types of files and the expected output.

Types of Files

Subscription fulfillment is a complex application. A magazine usually has different kinds of subscribers — those who've paid, those who haven't paid and were billed, and those who have paid their bills. We also have those who ordered bulk subscriptions to one address, a few who want the magazine sent first class, and some promotional copies.

Subscriptions come in every day and someone has to enter the data, proofread it, and merge it with the large subscription data base. At publication time, we have to print mailing labels in postal zone and zipcode order (second class bulk mailings have to be in postal zone order, and within each zone, in zipcode order).

We assumed we could manage up to 5,000 subscribers easily in a system using floppy disks. We *distributed* our *processing* by having a typist use an Osborne or Kaypro portable computer for data entry, while we used the larger system for editing, merging, combining files and producing output. We transferred data every day from the Osborne or Kaypro to the larger system (using Modem7 and a direct connection through serial ports).

We passed the 5,000 mark long ago and needed a hard disk to hold the data base. However, our file management was first designed to handle the application with eight-inch floppy disks on a standard CP/M system (our floppies hold up to 600K each in the double-density single-sided format, and our computer is the Alspa ACI-2, a standard two-drive system).

Obviously we had to invent a file naming scheme that would separate data base files into appropriate categories. This is extremely important, and should not be overlooked. For example, we use a names with the creation date appended to it for data entry files, to archive copies of them in an order that makes sense (by date of entry).

Early in the game we realized that if we ever wanted to *combine* separate files to make a large data base, the process would not be easy unless we decided on *one structure* that would fit every type of subscription file that could be conceivably combined.

On the other hand, reader service cards carry a lot of questionaire information per person, and each person usually wants information about several products. If we tried to use the same structure for both subscription fulfillment and reader service data, we would have wasted too much space in the subscription data base because dBASE II, like many other personal computer DBMS (data base management system) packages, allocates space for all fields in each record even if data exists in only some of the fields.

We put reader service card information into a data base with a different structure. However, as you will see, we used the *same field names* in both structures so that data could be easily transferred from one data base to the other. This is perhaps the most obvious advantage dBASE II has over other data base management systems: fields are named like variables, rather than numbered by the order they are stored.

Output

Subscriptions are not handled in one mailing. Some subscribers receive an issue only, some an issue and a bill in an envelope, some an issue and a renewal notice, and some an issue and a set of back issues or summary cards. Label sets have to be printed for each type of mailing.

In addition, we must produce different types of labels: one type for U.S. subscribers, one for foreign subscribers, one for back issue requests, and one for reader service card labels. We realized that if we used the same variable name in all our data base structures for last name, address, zipcode, etc., we could use *one label printing routine* for all data bases.

This is not a realization to be taken lightly. We've seen dBASE II applications that were more complex than they needed to be, simply because the variable names were different from each type of data base. Why not design one report form or command file to print a report or label for any data base you care to design? You can do this with dBASE II if you stick to the same variable names for the same kind of data.

The Structure

The actual sizes of fields and their placement within a record were decided upon after we experimented with the data entry process and with labeling. To do the experiments we needed a sample structure to work with. After guessing at things like the order of typed entries and the widths and lengths of labels, we arrived at a sample structure.

TALK WITH THE STARS
COMMX
HP125 1000
VAX
RBBS
PDP 11
UNIVAC
TELEVIDEO
COMPU-SERVE
DEC 10 20 VT180
TRS 80
THE SOURCE
GODBOUT
MORROW
KAYPRO
ZENITH
XEROX
ALTOS
APPLE Z80
TELEX TWX
SANYO
VECTOR GRAPHICS
CROMEMCO
IMS
OSBORNE
MONROE
IBM TSO
OTRONA
NORTH STAR
ITHACA

the best VALUE anywhere
COMMUNICATIONS SOFTWARE PACKAGE
COMMX-PAC™

- COMMX Smart Terminal and File Transfer Auto Dial and Logon Files for Easy Linkup Protocols for Timeshare, MODEM 7, CRC16 FORTRAN version available for Mainframes
- Unattended "Midnite" Electronic Mail Subsystem
- CONSOLX Remote Operation of Your Computer
- Bulletin Board System with Database Access
- Database File Management System Included
- Utilities for Data Encription, Compression, and More
- CPU License $150 Detailed Manual $25

Contact Your Local Dealer or
Call or Write For Free Brochure

HAWKEYE GRAFIX

23914 Mobile, Canoga Park, Ca 91307 • 213/348-7909 U.S.A

Circle number 9 on the reader service card.

To create the data base, we typed the dBASE II *Create* command:

```
          create ↩
        ENTER FILENAME: substruc↩

FIELD           NAME,TYPE,WIDTH,DECIMAL PLACES
001             last:name,c, 16↩
002             firstname,c, 12↩
003             title,c, 10↩
004             org:name,c,22↩
005             address,c,32↩
006             city,c, 12↩
007             state,c,3↩
008             zipcode,c, 10↩
009             country,c,8↩
010             source,c,4↩
011             sub:type,c,2↩
012             num:subs,n,3↩
013             start:sub,n,3↩
014             end:issue,n,3↩
015             amt:paid,n,6,2↩
016             amt:bill,n,6,2↩
017             amt:bilp,n,6,2↩
018             pay:date,c,8↩
```

```
019          flag:rec,l↵
020          comment,c,25↵
021          enterdate,c,8↵
022          lastupdat,c,8↵
023          ↵

    INPUT DATA NOW?y
```

At this point, if your dBASE II program is installed for your screen, you should see an input screen that looks like this:

```
RECORD 00001
LAST:NAME      :                :
FIRSTNAME      :            :
TITLE          :          :
ORG:NAME       :                      :
ADDRESS        :                                :
CITY           :            :
STATE          :   :
ZIPCODE        :          :
COUNTRY        :        :
SOURCE         :    :
SUB:TYPE       :  :
NUM:SUBS       :   :
START:SUB      :   :
END:ISSUE      :   :
AMT:PAID       :        :
AMT:BILL       :        :
AMT:BILP       :        :
PAY:DATE       :    :
FLAG:REC       : :
COMMENT        :                         :
ENTERDATE      :    :
LASTUPDAT      :    :
```

If your version of dBASE II is not installed for full screen features, you may only see the first field (LAST:NAME), and you won't be able to use many of the handy editing features of dBASE II; however, the program is still very useful. It is much better to install the full screen features and get full use of the editing features, which resemble WordStar®.

Following each field name is a blank space ending with a colon, which shows the amount of space — the number of characters you can type before you fill it. You can type less than the full number, but you can't type more than the allowed number of characters.

Press RETURN to end fields with less than the full number of characters, or fill the field with characters to move to the next field. Fields and areas within fields left blank are filled with spaces.

When you finish record 1, press RETURN at the top of record 2 to stop entering data:

```
RECORD 00001
LAST:NAME        :  Bove↵
FIRSTNAME        :  Tony↵
TITLE            :  Edit & Pub
ORG:NAME         :  User's Guide to CP/M↵
ADDRESS          :  P.O. Box 3050↵
CITY             :  Stanford↵
STATE            :  CA↵
ZIPCODE          :  94305↵
COUNTRY          :  USA↵
SOURCE           :  aaa↵
SUB:TYPE         :  p↵
NUM:SUBS         :  1↵
START:SUB        :  2↵
END:ISSUE        :  7↵
AMT:PAID         :  18↵
AMT:BILL         :  ↵
AMT:BILP         :  ↵
PAY:DATE         :  03/21/83
FLAG:REC         :  y
COMMENT          :  comp#1, paid for 2-7↵
ENTERDATE        :  ↵
LASTUPDAT        :  ↵

    RECORD 00002
LAST:NAME        :  ↵
```

Press RETURN for any blank field, as shown. Numeric fields with decimals are automatically set to zero (e.g., 0.00) when you press RETURN to skip them.

When typing less characters into a field than the maximum, press RETURN to enter the characters. If you type the full amount, dBASE II moves you to the next field and you should *not* type RETURN.

If your keyboard is equipped with a "bell" or "beep," dBASE II uses it to tell you that you typed the full amount of characters in a field (TITLE in the example above). If you don't like this "beep" you can turn it off with this command:

```
    .set bell off↵
```

To display the structure of the data base file in use, type this command:

```
.display structure↵

STRUCTURE FOR FILE:  SUBSTRUC.DBF
NUMBER OF RECORDS:  00001
DATE OF LAST UPDATE:  00/00/00
PRIMARY USE DATABASE

FLD     NAME          TYPE    WIDTH   DEC
001     LAST:NAME     C       016
002     FIRSTNAME     C       012
003     TITLE         C       010
004     ORG:NAME      C       022
005     ADDRESS       C       032
006     CITY          C       012
007     STATE         C       003
008     ZIPCODE       C       010
009     COUNTRY       C       008
010     SOURCE        C       004
```

```
011    SUB:TYPE     C    002
012    NUM:SUBS     N    003
013    START:SUB    N    003
014    END:ISSUE    N    003
015    AMT:PAID     N    006    002
016    AMT:BILL     N    006    002
017    AMT:BILP     N    006    002
018    PAY:DATE     C    008
019    FLAG:REC     C    001
020    COMMENT      C    025
021    ENTERDATE    C    008
022    LASTUPDAT    C    008

** TOTAL **              00209
```

Automated Data Entry

Entering subscription orders is a task to be done every day, and the computer has to be accessible on an on-going basis. We decided to set up a portable Kaypro and a portable Osborne for data entry, and use the larger systems for processing, merging, accounting, producing reports, producing labels, etc.

Data entry requires the full dBASE II program, which takes up one of the two 5.25-inch disks you can use at any one time. The other disk is used for the data files.

For editing and processing, we use a new file every day; when we have more than one typist, each typist uses a different file. The result is a collection of small files, which are easier to edit. Once edited they become part of the large data base.

The files are named according to the type of subscription, with the date of entry appended to it (e.g., PAID07/21, BILL06/24, etc.).

Because a new file must be started every day, and the typist must take breaks and periodically copy the disk, we developed two automatic data entry operations: *startup* and *continue*. We present the simpler *continue* operation first.

The CONTINUE Command File

We use *command files* to control dBASE II operations. A command file is a text file with dBASE II commands as they would be typed on the dBASE II command line. To run a command file, you type *DO* followed by the name of the command file. Command files have the ".CMD" filename extension.

You can perform data entry simply by typing the *APPEND* command; however, we control data entry with two command files: one called STARTUP.CMD for the beginning of a day, and one called CONTINUE.CMD for continuing entry into the same daily file after taking breaks.

The goal for CONTINUE.CMD is to allow a typist to start dBASE II after taking a break, and continue entering data into the same file. Here's how CONTINUE.CMD is used:

```
. do continue↵
Type name of data base: paid07/21↵
```

(The next record of the data base appears on the screen.)

To create the CONTINUE.CMD command file, type:

```
. modify command↵
ENTER FILENAME: continue↵
```

If your dBASE program is installed with screen editing, your screen should clear; otherwise, a blank line should appear. You can now type commands as they would appear on the dBASE II command line. The CONTINUE.CMD command file looks like this:

```
set bell off↵
accept 'Type name of data base' to filename↵
use a:&filename↵
append↵
```

This command file starts by setting the "bell" (keyboard beep) off (the beep tone may be annoying). The *SET* command can turn many features on or off.

Next it uses the *ACCEPT* command, with the message *'type name of data base'*, to ask the typist for the name of the file. We assume the typist will type the name for a file already started that day (to start a new file, the typist must use the command *DO STARTUP*).

The CONTINUE.CMD file contains a *USE* command to open the data base file. The *USE filename* command is the only way to "use" an existing data base. The *filename* is the name of the data base file. Data base filenames are assumed to have the ".DBF" extension, so you don't have to type it; however, if you don't use a ".DBF" extension, you must type the extension you use. The *CREATE* command creates a data base file with the ".DBF" extension.

The character string *&filename* is a *variable* much like a MailMerge variable (MailMerge is an added option to WordStar). It holds whatever was typed in response to the *ACCEPT* command, and substitutes the contents for the string *&filename*, joining it with the *a:* to form the complete filename (since data files are located on drive A).

The last line of CONTINUE.CMD is the *APPEND* command, used to display the next blank record and to start data entry.

The STARTUP Command File

The STARTUP.CMD command file (more complicated than the CONTINUE.CMD command file) lets a typist create a new file on the Kaypro data disk (used in drive A, with dBASE program files on drive B) without appending to or demolishing an existing file. If the typist types the name of an existing file, it displays a message telling the typist to use *DO CONTINUE*:

```
. do startup↵
Type name of data base: paid07/21↵
Is paid07/21 right? Or should you use DO CONTINUE?
```

ROUTINE CANCELED

```
. do startup↵
Type name of data base: paid07/22↵
```

(First record is displayed for data entry.)

To create the STARTUP.CMD command file, type the following into a text file named STARTUP.CMD:

```
set bell off↵
accept 'Type name of data base' to filename↵
if file('a:&filename')↵
    ?'Is &filename right? Or should you use DO CONTINUE?'↵
    cancel↵
else↵
    use substruc↵
    copy struc to a:&filename↵
    use a:&filename↵
    append↵
endif↵
```

This command file shows how you can organize dBASE II commands. We use the *IF*, *ELSE* and *ENDIF* keywords to make a decision.

The logical if-then-else expression starts with the *FILE(filename)* function to determine if the *filename* already exists:

```
if file('a:&filename')
   ?'Is &filename right? Or should you use DO CONTINUE?'
   cancel
else
```

If the filename exists, dBASE performs the next two statements. The first statement starts with a question mark (*?*), which is shorthand for the *DISPLAY OFF* command. The *DISPLAY OFF* command displays fields without the record number. The message to be displayed must be within quotation marks.

The second statement (*CANCEL*) cancels the command file operation.

If the filename does *not* exist, dBASE performs whatever follows the *ELSE* keyword up to the *ENDIF* keyword:

```
else
    use substruc
    copy struc to a:&filename
    use a:&filename
    append
endif
```

The first statement puts the first file we created, SUBSTRUC.DBF, into use as the "model" of the file structure. The second statement *copies* that model to a new file on drive A — the file whose name is typed and received in the variable *&filename*.

The third statement puts this new file into use as the data base, and the fourth statement (*append*) displays the first record for data entry.

Stopping Data Entry

The *endif* statement marks the end of the if-then-else expression. Since there are no more statements, it also serves as the last command. Data entry continues until break time, when the typist presses a RETURN for the first field of a new blank record, and returns to the dBASE II command line to type this final command:

```
. quit↵
```

This is the proper way to end dBASE II operations and return to CP/M.

At this point, the typist makes a backup copy on another disk providing us with two identical data entry disks with data files in the "as-they-were-typed" stage. We call these disks our *data entry archive*.

With copies stored in the archive, we transfer the data entry files to eight-inch disks on the Alspa system, to prepare them for editing. After editing, we transfer copies of the corrected files back to the archive disks. Then we merge the edited data entry files into a single data base file using a version of the *APPEND* command, described later.

Data Entry Fields

Doing the data entry showed us that typists like to start with the information they know first: the last name, followed by first name, followed by title and company, address, state and zip, *then* information about the subscription.

We also noticed that typists might not be sure the information is correct, and so there needs to be a way to mark the record. We added a "flag the record" field and a comment field based on our experience.

In programs other than dBASE II we might be forced to define our *key* fields — the ones we would use for sorting, such as zipcode as a primary key, or last name (primary key) and first name (secondary key), etc.

dBASE II gives us the flexibility to use any field as a key to build an index to the data. To sort by zipcode, we only have to create an index file using the zipcode field as the key. Index files are short and take much less time than actual sorting of the data base.

Knowing this, we designed a structure with fields in the order that they're typed, and with enough separate fields to use as indexes for a variety of tasks.

Displaying and Listing Data

A remarkable feature of dBASE II is the ability to select records for displaying, editing or printing. Two dBASE commands you will use constantly are *LIST* and *DISPLAY*. *DISPLAY* by itself displays the record currently pointed to by dBASE:

```
. display↵
00001 Bove    Tony    Edit & Pub   User's Guide Magazine
P.O.  Box 3050               Stanford  CA   94305      USA    aaa  p  1  2
7   18.00    0.00    0.00  03/21/83  .T. comp#1, paid for 2-7
```

The record number is 00001 (the highest you can have in one file is 65535). The data fields are displayed one after another on as few lines as possible.

DISPLAY displays one record at a time unless you specify a *scope expression* such as *ALL*, or a *FOR* expression that causes dBASE to search for records to fit the criterion:

```
. display all for last:name= 'B'↵
```

This command will display all records whose LAST:NAME field starts with the letter *B*.

LIST performs the same function as a *DISPLAY ALL* command, and is better suited for long lists and printouts (the *DISPLAY* command pauses every 15 records and waits for you to press a key, which is fine for reading on the screen, but not for printing).

If you type *LIST* by itself, the command lists every record in the data base. With the printer on, you can type these two commands to print every record:

```
. set print on↵
. list last:name org:name city zipcode start:sub num:subs↵
```

LIST and *DISPLAY* can be limited to displaying only the fields you specify. For example, we could display only last names and zipcodes for records whose last names begin with *B*:

```
. display all last:name, zipcode for last:name= 'B'↵
```

To move to another record and *DISPLAY* it, use the *GOTO* command with the record number:

```
. goto 25↵
. display↵
```

The *SKIP* command moves you to the next record:

```
. skip↵
. display↵       — displays record 26.
```

Other *GOTO* commands move you to top and bottom of the file:

```
. goto top↵
. display↵       — displays record 1.

. goto bottom↵
. display↵       — last record in the file.
```

Other ways to see data include the *EDIT* and *BROWSE* commands.

Editing Records

EDIT lets you select a record by number to change the data in its fields. Type *EDIT 1↵* to edit record number one, and the record appears on the screen. dBASE provides the simple WordStar Control commands that let you move the cursor to any field or within a field to change or delete the data.

Use the familiar Control-*X* or Control-*E* commands to move up or down by field, and Control-*S* or Control-*D* to move left or right within a field. Type over text to change it, or use Control-*G* to delete characters or Control-*Y* to delete all data in the field. Numeric fields are changed by typing the number as you would for a SuperCalc cell. Move to the field, type the new number and press RETURN.

WordStar users: *don't use* Control-*Q* as you would in WordStar to move quickly to the beginning or end. Control-*Q* performs an "abandon" which returns you to the dBASE II command line without saving your edits for that record. (This is a radical departure from the general WordStar similarity).

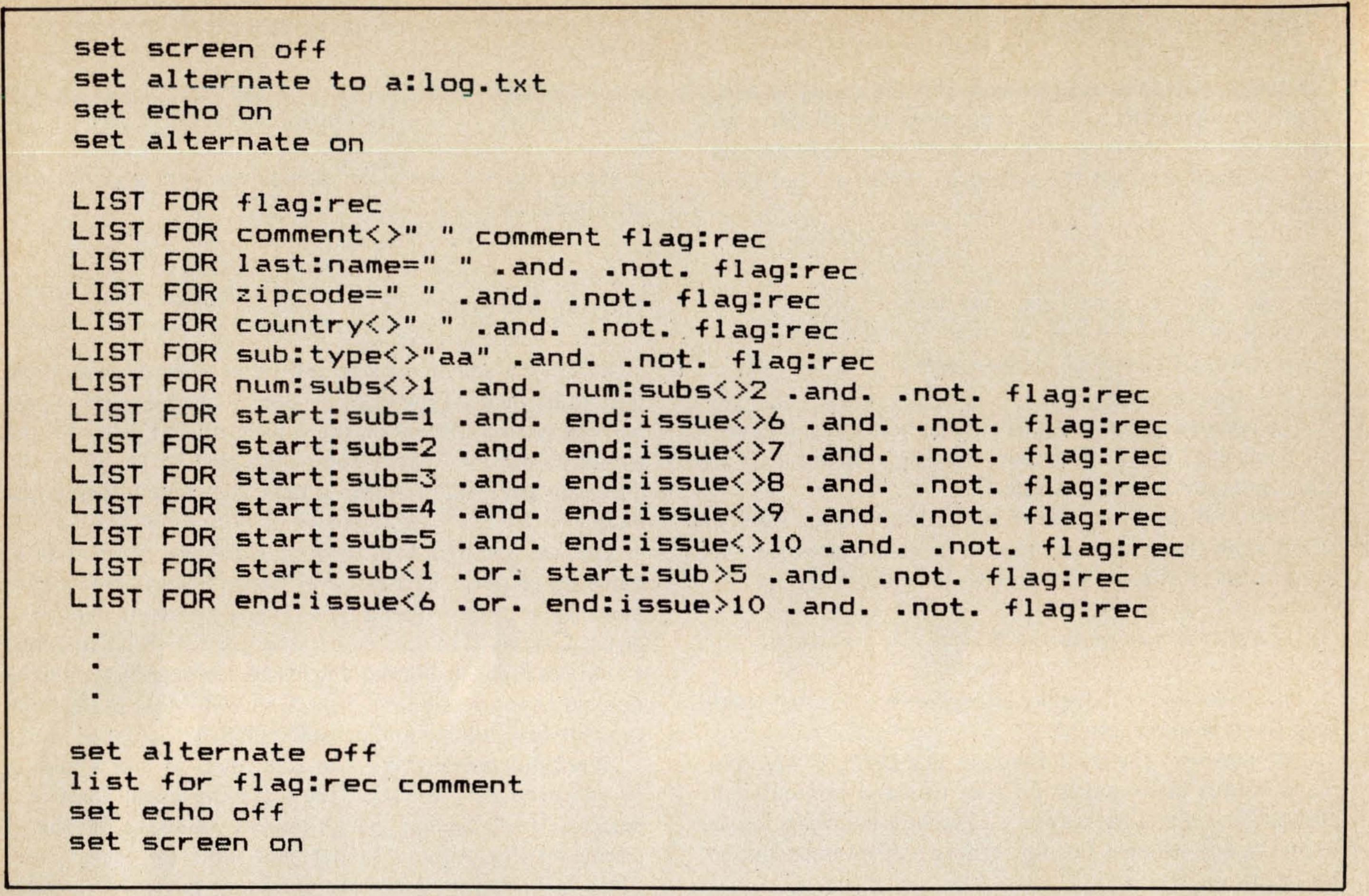
To move forward one record, use Control-*C*, and to move back one record, use Control-*R*. Both commands save the edits you made to the current record.

To save the edits and leave edit mode, type Control-*W* (another radical departure from WordStar). To quit without saving the edits of the most recently edited record, type Control-*Q*.

You can scan records quickly in *EDIT* mode to spot misplaced data. (Many mistakes involve data typed into the wrong fields, and a record *EDIT* takes care of the problem.) Type Control-*C* commands to move forward in the file (our version can move very quickly from record to record).

Here's a summary of the *EDIT* mode Control-key commands:

↑*X*	Move the cursor one field forward (one line down).
↑*E*	Move the cursor one field back (one line up).
↑*D*	Move the cursor one space forward within a field.
↑*S*	Move the cursor one space back within a field.
↑*Y*	Erase the field's contents.
↑*V*	Turn on or off the insert mode to insert characters.
↑*G*	Erase one character at the cursor position.
↑*W*	Save the changes you have made.
↑*Q*	Ignore the changes you made (the record is unchanged).
↑*U*	Mark the record for later deletion from the data base.
↑*C*	Move forward one record.
↑*R*	Move backward one record.



```
set screen off
set alternate to a:log.txt
set echo on
set alternate on

LIST FOR flag:rec
LIST FOR comment<>" " comment flag:rec
LIST FOR last:name=" " .and. .not. flag:rec
LIST FOR zipcode=" " .and. .not. flag:rec
LIST FOR country<>" " .and. .not. flag:rec
LIST FOR sub:type<>"aa" .and. .not. flag:rec
LIST FOR num:subs<>1 .and. num:subs<>2 .and. .not. flag:rec
LIST FOR start:sub=1 .and. end:issue<>6 .and. .not. flag:rec
LIST FOR start:sub=2 .and. end:issue<>7 .and. .not. flag:rec
LIST FOR start:sub=3 .and. end:issue<>8 .and. .not. flag:rec
LIST FOR start:sub=4 .and. end:issue<>9 .and. .not. flag:rec
LIST FOR start:sub=5 .and. end:issue<>10 .and. .not. flag:rec
LIST FOR start:sub<1 .or. start:sub>5 .and. .not. flag:rec
LIST FOR end:issue<6 .or. end:issue>10 .and. .not. flag:rec
 .
 .
 .

set alternate off
list for flag:rec comment
set echo off
set screen on
```

Figure 1. *The EDITPAID.CMD command file that lists records thought to have data entry typos or other mistakes, used as a "filter" throughout the file. The commands do not change data, they simply list records to a "log" file and display them on the screen.*

Automatic Proofreading

A data base is usually less accurate than the paper files that generated it, if only because the typists make mistakes. Although typists can correct their typing while entering data into a record in *APPEND* mode, you have to invent your own data entry routine to correct mistakes. The only foolproof method is to painstakingly proofread each record and compare it to the paper files. dBASE II provides the *EDIT* mode for making changes to each record.

However, if time is more important than total accuracy (e.g, you can tolerate a few misspellings until you have time to look at printouts), design a proofreading routine to catch the most typical mistakes, and use such a routine as a filter for all newly-entered data.

The command file in figure 1 uses *FOR* expressions to compare data in different fields, with *.AND.*, *.OR.*, *.NOT.* and the logical greater-than, less-than or equal-to symbols to form logical comparisons. (You must use periods around the words *.AND.*, *.OR.* and *.NOT.*).

The comparisons may yield accurate as well as inaccurate records, depending on how narrow you focus them. Use comparisons that allow you to form an "opinion" of the record — and if there's slight doubt, check it out.

Here's a sample comparison: if a subscriber's first issue is no. 5, the last before renewal should be no. 10 (six issues). This can be expressed as:

```
list for start:sub= 5 .and. end:issue<> 10↩
```

The *FOR* keyword starts the phrase. *.AND.* joins the two conditions so that the phrase is true *only if* both conditions are true: if the starting issue of the subscription is *equal to* five, *and* the ending issue is *not equal to* ten.

The scope is narrow enough to flag the obvious typos or misplaced data. It is also too wide and catches all subscribers who ordered two years rather than one.

LIST displays the record number and the entire record on the screen, so that you can spotcheck the data to see if it is correct. If something is unusual, remember the record number and use *EDIT* to look at the record.

Figure 1 shows part of a command file called EDITPAID.CMD, used to "filter" data entry records for our paid subscriber data base. The command file does not change or mark records — it simply displays the records that match each criterion. The display is also copied to a disk file (LOG.TXT) for viewing or printing. The display includes the entire record, but the record number is all that is needed to select a record in *EDIT* mode.

EDITPAID.CMD uses variables shown in the *STRUC-*

TURE and data entry examples. The body of the command file is explained here and is followed by sections that explain the "preamble" and "postscript" areas.

Each command lists records that satisfy the logical *FOR* expression. The first *LIST* command lists all records that have a "Y" or "y" in the flag:rec field:

```
list for flag:rec↩
```

This command catches all records we "flagged" during our quick edit session (we typed *Y* or *y* in this field and wrote a comment in the comment field). We "flag" records that are suspicious or blatantly wrong.

The second *LIST* command lists records whose comment fields are *not equal* (⟨⟩) to a blank field:

```
list for comment⟨⟩' ' comment flag:rec↩
```

The above command lists only the record number and comment field for each record flagged. This is a useful screen display for referring to specific comments.

Subsequent *LIST* commands list records that were not yet "flagged," which is why each expression ends with *.NOT. flag:rec.*

The "Preamble" and "Postscript" sections of the command file are *SET* commands that control the screen and printer:

```
set screen off↩
set alternate to a:log.txt↩
set echo on↩
set alternate on↩
.
.
.
set alternate off↩
set echo off↩
set screen on↩
```

The first *SET* command turns full screen mode off, so that displays will scroll without erasing the screen.

The second command sets the *alternate file* to A:LOG.TXT. The alternate file works with the *SET ALTERNATE ON* command to copy all screen and keyboard activity to a disk file. All results of the *LIST* commands are stored in A:LOG.TXT, where they can be examined using any text editor or dBASE II's *MODIFY COMMAND* command, or LOG.TXT can be printed.

The third *SET* command sets echo mode to on, causing the commands in EDITPAID.CMD to be displayed on the screen as they are executed. The display of commands coincides with the results of the commands, showing us which commands caused which records to be displayed.

The fourth *SET* command sets alternate mode on, causing all subsequent screen and keyboard activity to be copied to the alternate file, which is set to be A:LOG.TXT.

The final three *SET* commands set everything back to the way it was before starting the editing operation. They turn off echo and alternate modes and turn on full screen mode for normal operation.

Consolidating and Archiving

In small applications (less than 3000 names), you could merge all data entry files into one data base file with 3000 records on a floppy disk. For larger applications you need to separate records into several data base files, in order to archive them properly. A data base file cannot be larger than the amount of space provided on a data base archive disk.

With a collection of data base files forming one logical data base, it pays to have a good file naming scheme that subdivides the data base into logical sections.

We store the subscription data base in separate archive files relating to the type of subscription and its starting issue:

Filename	Type of Subscription
PD/1	Paid subscriptions starting with issue 1.
PD/2	Paid subscriptions starting with issue 2.
BL/1	Billed subscriptions starting with issue 1.
BL/2	Billed subscriptions starting with issue 2.
.	
.	
.	

By separating the data base into these categories, we can easily select the appropriate subscribers for renewal promotions, bills, bill reminders, etc. To make these data base files, we must first merge the data entry files into a temporary file, and select the appropriate records for each data base file. The *APPEND FROM FILE* command will append an entire data base file to another, effectively merging the two files:

```
. use pd/1↩
. append from paid7/21 for start:sub= 1↩
```

To prepare for a mailing, we combine all PD, BL and other files into large files containing all names to be sent mailings.

We use generic filenames for these files: PAID (paid in advance), BILL (billed), BILP (billed subscribers who've paid), SPEC (specials and bulk orders), 1STP (1st class subscribers who've paid), 1STB (1st class who are to be billed), etc.

These files are subdivided into smaller files, one for each postal zone. A zipcode index is prepared for each zone file, and labels are produced in zip order within each zone. dBASE II makes this process easy with its ability to generate indexes.

Preparing an Index

A dBASE II data base can take forever to sort. The alternative, which is easier, faster, and a more efficient use of your time and the computer's time, is to *make an index* to the file.

You can make an index using *any field as a key*. The key fields in the index are sorted into ascending order by character, and they make the data base appear to be sorted for all future operations while the file is in use. The data base itself is not changed.

To prepare an index in alphabetical order by last name, type this command:

```
. index on last:name to paid/1x↩
```

This example creates an index file named PAID/1X.NDX. If you now do a *GOTO TOP* command and then a *LIST* command, the file will be displayed in alphabetical order.

Once an index is created, you can use the index with the data file at a later date by specifying the index file with the data file in the *USE* command:

```
. use paid/1 index paid/1x↩
```

The data base PAID/1.DBF, with its index file PAID/1X.NDX in use with it, appears to be sorted in alphabetical order but is actually not sorted — the index file does the work. If you use the *APPEND* command to add data, the index file will keep the new data in alphabetical order automatically. Any *EDIT* mode changes also update the index file, as do record deletions.

We keep an index file by last name on the disk with the data base since it takes up very little space compared to the data file. We also keep an index by zipcode order:

```
. index on zipcode to paid/1z↩
```

With the index file PAID/1Z in effect and in use with PAID/1, we can print a label for each record in PAID/1 in zipcode order. Of course, if we updated PAID/1 with address changes, and used the PAID/1X index file during the updating, the PAID/1Z file would not contain the updated zipcode. We have to make a new zipcode index, which does not take much time. Only one index can be in use with one primary file (alternate files discussed in another installment).

Making a new index takes much less time than sorting, and it doesn't change the data base, nor does it require much room on the disk. Sorting comes in handy for other applications discussed in other installments.

Printing Records

dBASE II provides three ways to print information: defining and using a report, writing a command file containing statements (described in another installment), or duplicating screen activity to the printer.

The first method is the best for summary data, for setting up columns, and for performing record-by-record arithmetic. We found it inadequate for performing two functions: printing the entire file by last name in alphabetical order, and printing labels.

The second method, using statements, is similar to writing BASIC programs with PRINT USING statements. It is tedious and tricky, but ultimately the most accurate method of printing data where you want it. This method requires use of command files.

The third method is to use the *SET PRINT ON* command and print everything that happens next on your screen. One well-phrased *LIST* command might do the job:

```
. set print on↩
. list last:name org:name city zipcode start:sub num:subs↩
```

Listing By Last Name

If the last name index is in use with the file, the above fields of each record should appear on your screen *and* on your printer in alphabetical order. This might be enough unique information about each person to make a useful listing.

For complete listings with totals, we use a command file that takes each person's amount paid and number of subscriptions (numeric fields), stores them in temporary variables, and prints their sums at the end of the file.

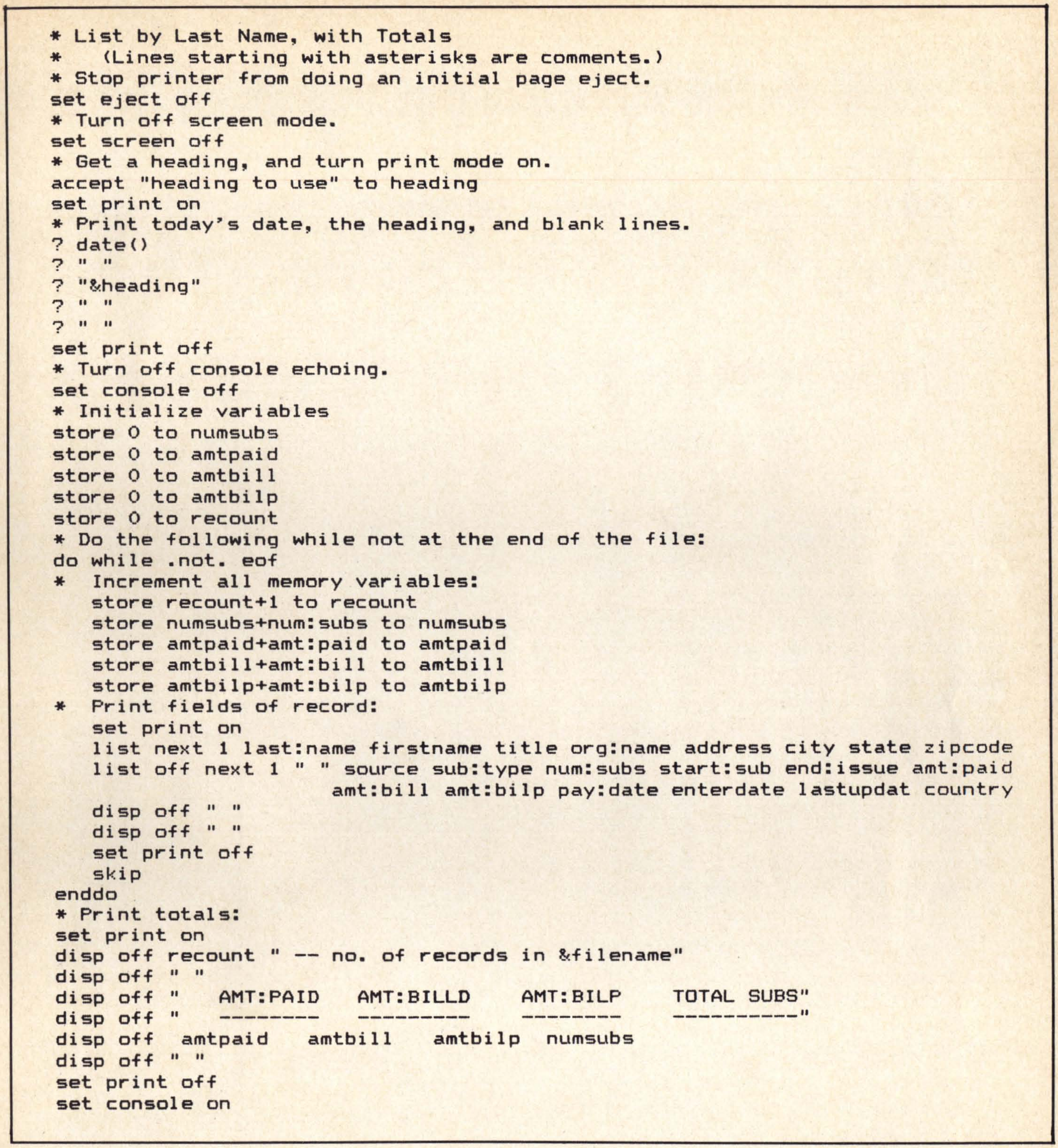
For dBASE II buffs we provide the command file in figure 2, with comments. For beginners, we promise more explanations in future installments (dBASE II is a very large package).

Mailing Labels By Zip

If the zipcode index file is in use with the data base, the following set of commands in a command file will print the appropriate fields of each record on single-sheet ("one-up") labels:

```
.
.
.
do while .not. eof
set print on↩
disp '&filename' start:sub end:issue sub:type ↩
disp off firstname last:name num:subs↩
disp off org:name title↩
disp off address↩
disp off city state ' ' zipcode↩
? ' '↩
skip↩
enddo↩
set print off↩
.
.
.
```

In this example, the name of the data base (assigned to the variable *&filename* elsewhere in the command file) is printed at the top of each label. The first *DISPLAY* command displays the record number and the filename of the record, plus some



```
* List by Last Name, with Totals
*   (Lines starting with asterisks are comments.)
* Stop printer from doing an initial page eject.
set eject off
* Turn off screen mode.
set screen off
* Get a heading, and turn print mode on.
accept "heading to use" to heading
set print on
* Print today's date, the heading, and blank lines.
? date()
? " "
? "&heading"
? " "
? " "
set print off
* Turn off console echoing.
set console off
* Initialize variables
store 0 to numsubs
store 0 to amtpaid
store 0 to amtbill
store 0 to amtbilp
store 0 to recount
* Do the following while not at the end of the file:
do while .not. eof
*   Increment all memory variables:
    store recount+1 to recount
    store numsubs+num:subs to numsubs
    store amtpaid+amt:paid to amtpaid
    store amtbill+amt:bill to amtbill
    store amtbilp+amt:bilp to amtbilp
*   Print fields of record:
    set print on
    list next 1 last:name firstname title org:name address city state zipcode
    list off next 1 " " source sub:type num:subs start:sub end:issue amt:paid
                       amt:bill amt:bilp pay:date enterdate lastupdat country
    disp off " "
    disp off " "
    set print off
    skip
enddo
* Print totals:
set print on
disp off recount " -- no. of records in &filename"
disp off " "
disp off "   AMT:PAID   AMT:BILLD    AMT:BILP    TOTAL SUBS"
disp off "   --------   ---------    --------    ----------"
disp off  amtpaid   amtbill   amtbilp  numsubs
disp off " "
set print off
set console on
```

Figure 2. *This command file lists each record in the data base, adding numeric values from each record and providing sums of those fields at the end of the file.*

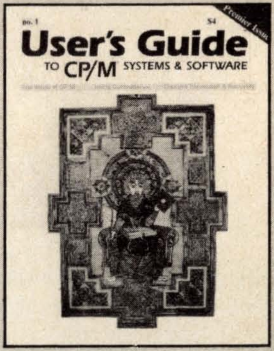
other information, on the top line of each label for easy tracking. The other *DISPLAY OFF* commands leave out the record number and just display the data.

End Note

This installment started with creating a structure, entering new data into the data base, displaying information in the data base, editing records, writing command files, and printing listings and mailing labels. We used "crude" command files in this installment; in the next, we'll show how to link command files together (structured programming) and develop menu-driven applications. Also, more on printed output, preparing reports, and gathering statistics from multiple data bases.

"At last, a magazine with practical, hands-on information for computer users."

— letter to the editors

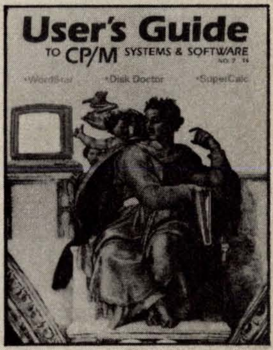

no. 1
$4
Premier Issue
User's Guide
TO CP/M SYSTEMS & SOFTWARE

User's Guide™ *is for those who don't care about bits and bytes, but need information on how to make their computers more useful.*

User's Guide *introduces you to the world of CP/M®, the most popular type of small computer system.* **User's Guide** *provides in-depth coverage of CP/M systems and software.*

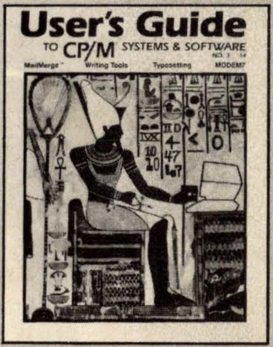
For example:

- *How to get more mileage out of popular CP/M programs like WordStar®, SuperCalc®, and dBASE II®.*
- *How to gain access to thousands of* **free** *programs in the public domain using a program called MODEM7.*
- *How to recover data from a damaged disk.*
- *How to set up and use a small business accounting system.*


User's Guide
TO CP/M SYSTEMS & SOFTWARE
NO. 2
•WordStar
•Disk Doctor
•SuperCalc

User's Guide *features well written articles and clear, step-by-step instructions that help you get the most out of your computer investment.*

For a one year subscription (six issues), use the card next to this ad, or send a check for $18.00 to **User's Guide**, *P.O. Box 3050, Stanford, CA 94305. Foreign subscriptions: $24 pre-paid.*


User's Guide
TO CP/M SYSTEMS & SOFTWARE
NO. 3
MailMerge
Writing Tools
Typesetting
MODEM7

User's Guide
TO CP/M SYSTEMS & SOFTWARE

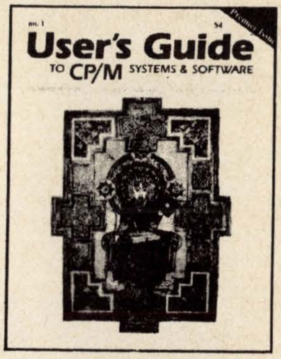
CP/M is a trademark of Digital Research.
WordStar is a trademark of MicroPro International.
SuperCalc is a trademark of SORCIM Corp.
dBASE II is a trademark of Ashton-Tate.
User's Guide is a trademark of TUG, Inc.

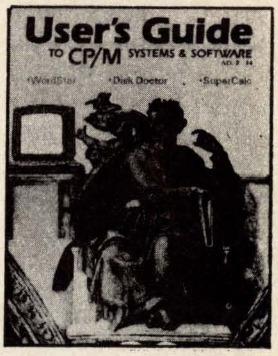
FREE INFORMATION
User's Guide Reader Service Card

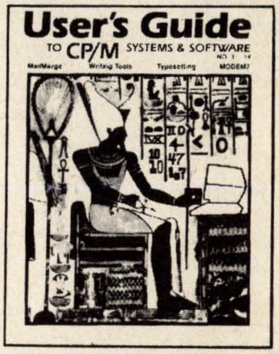
User's Guide
P.O. BOX 3050
STANFORD, CA 94305

Each advertisement and product announcement in this issue has a reader service number. To receive by mail from the manufacturer, simply circle the same number on this reader service card, and drop the card in the mail. Your name and address is sent on to the appropriate manufacturer, who sends the information.

The service helps you get detailed information about products you may want to purchase. *User's Guide* does not sell or recommend products associated with this service.







1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

Please take the time to fill out:

1. How would you classify your experience with computers?
   ☐ Novice ☐ Intermediate ☐ Advanced
2. Which computer do you use?__________
3. Please describe your system.__________
   Memory __________
   Printer __________
   Monitor __________
   Storage devices __________
   Other Components __________
4. Name some applications programs that you use most with your computer.
   ____
5. Where do you use your computer?
   ☐ Home ☐ School ☐ Small-large business environment
   ☐ Large-business/Corporate environment ☐ Other __________

Name __________ Age _____ Sex _____
Occupation __________ Company __________ Title __________
Address __________ City/State/Zip __________

Mail To: P. O. Box 3050, Stanford, CA 94305

5

Advertiser Index

Service		Page
1.	Alspa	C4
2.	Centaur	11
3.	Computer Thaumaturgy	55
4.	Computing!	C3
5.	Data Access Corp.	65
6.	Digital Marketing	53
7.	EWDP Software	11
8.	First Osborne Group	49
9.	Hawkeye Grafix	33
10.	Heritage Software, Inc.	9
11.	International Software Alliance	19
12.	OmniSystems International	9
13.	Pickles & Trout	59
14.	Professional Aids Company	23
15.	The Systems Group	71
16.	T/Maker Company	3
17.	Techne Software Corp.	5
18.	User Friendly Software	19
19.	Wizard of OsZ	83
20.	The Word	C2

Perfect Filer

A Data Base Program Bundled With Kaypro Systems, and Available for CP/M Systems and the IBM PC

by Matthew McClure

My introduction to Perfect Filer was brief. It was the first day of my new job at the *Whole Earth Software Catalog* and *Review*, and Art Kleiner (Editor-at-Large) assigned me the task of designing and maintaining a data base of potential reviewers. We planned to send writer's guidelines to potential reviewers in a week or ten days, and Perfect Filer was going to produce the mailing labels.

I wanted to get started quickly, so when Art showed me the KayPro 2 and gave me a few floppy disks and a stack of magazines, tabloids and books, including the Perfect Filer manual, I did not take the time to read the manual from cover to cover.

I am familiar with file maintenance programs and helped write such a program for the Commodore 64. A glance at the manual showed that Perfect Filer could support my needs: design the entry form, enter the data, make changes to the data, select which records to print, sort the records according to fields, and format the reports to be printed.

Perfect Filer allows only one file per disk; fortunately, that was just what I needed. I was looking forward to using my first CP/M machine, and my initial glance at the manual indicated that getting this project going would be easy, perhaps even fun.

Getting Started

Art had given me four disks: a CP/M system disk, a Perfect Filer program disk, and two data disks: one for "work" and one for "backup." I put the Perfect Filer program disk in drive A and the data disk in drive B and pressed the reset button. The screen displayed the KayPro CP/M sign on message and the A> prompt.

I typed:

```
A>FILER⏎
```

The program asked for the drive containing my data disk, so I typed a *B*. I was rewarded with the disappointing message, Cannot open db.def file.

I decided the message was a cue to define my database first, so I pressed the reset button and reached for the manual, where I quickly discovered the CREATE program and typed:

```
A>CREATE⏎
```

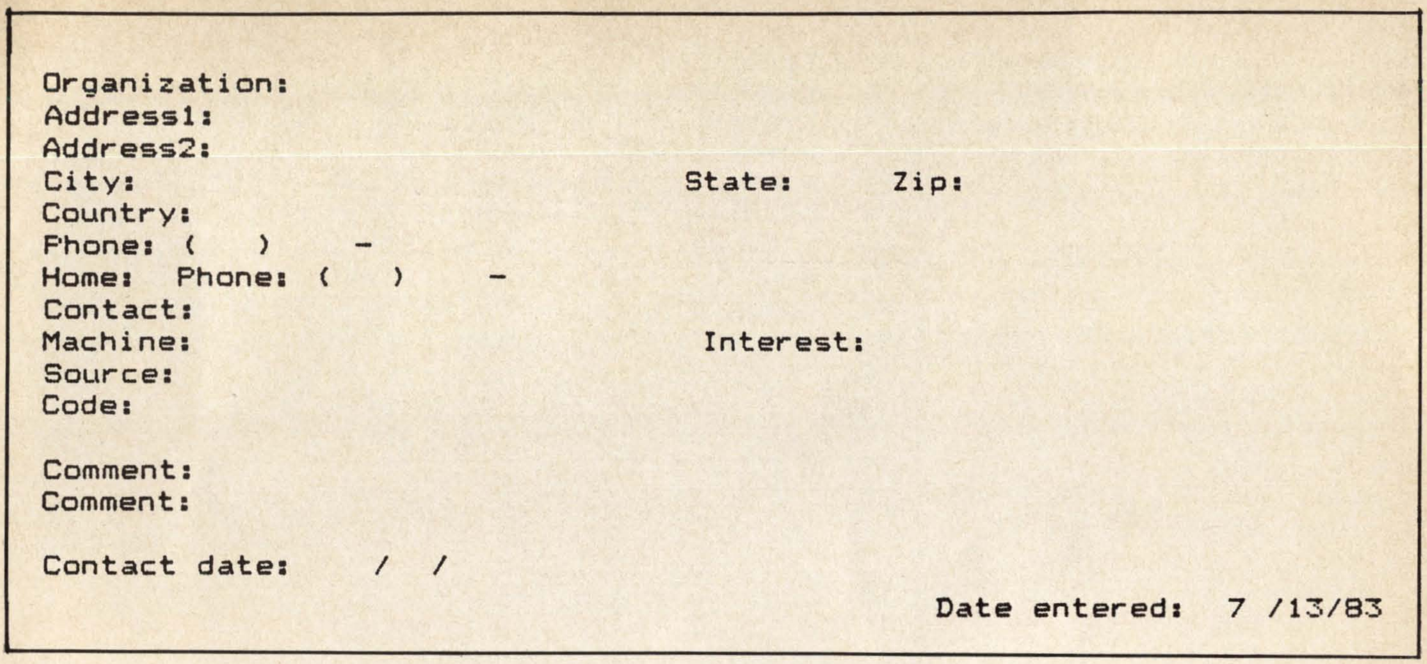
Perfect Filer asked for the drive with the data disk, so I typed *B*. This time I was rewarded with a menu headed Data Base Generator.

I wanted the first choice in the menu, Create new data base, which had an X in front of it. I pressed RETURN and the X moved down to the next choice, Modify existing data base. Oops.

I tried to make the X move back up using the up and down arrow keys (they didn't do it) and several others, but I didn't try the backspace key (which was the correct choice) until later.

I tried typing an *X* to see if that would indicate my choice. Sure enough, the screen cleared and an error message appeared on the screen: Cannot open b:db.def for reading. I pressed reset again, typed *CREATE* and *B*, got the same menu, and this time was careful to type an *X* while the cursor was still in [X] Create new data base.

As a beginner I had nothing to lose by this experimenting. Commercial software products (of quality) are typically tested this way for quality assurance before they are offered for sale to



```
Organization:
Address1:
Address2:
City:                              State:     Zip:
Country:
Phone: (   )    -
Home: Phone: (    )    -
Contact:
Machine:                            Interest:
Source:
Code:

Comment:
Comment:

Contact date:    /  /
                                               Date entered:  7 /13/83
```

Figure 1. *A data entry screen, with fields and prompt messages, created by Perfect Filer.*

the public, so I wasn't expecting my disk to get its data mangled.

I am a confirmed member of the when-all-else-fails, read-the-directions school of computing. As Bucky Fuller pointed out, we often learn best by making mistakes. I confess to having learned a lot working with Perfect Filer.

Another menu appeared, and again the top choice was the one I wanted: [ ] Design screen layout. I pressed another *X*, and the screen went blank, with the cursor in the upper left-hand corner. I had a moment's feeling of insecurity: I was just getting used to those nice, friendly menus with their Xs. Now I was on my own.

Setting Up Data Entry: Templates and Field Tags

Someone in the office had mentioned that Perfect Filer had something called *templates*, which sounded interesting. I decided to look them up in the manual. Three of them immediately struck me as useful: the organization, phone and date templates.

A *template* is a collection of fields that go together; for example, a date template is made up of the month field, the day field and the year field; a telephone template puts the area code in one field and the number in another.

Setting up the screen turned out to be pretty easy. I typed the prompt message for each field in the place I wanted it on the screen, followed by either a Control-*I* to define it as a single field, or Control-*T* for a template.

My first command, at the upper left-hand corner of the screen, was a Control-*T*, which put me back to a comfortable menu with Xs in it. The menu let me choose the kind of template I wanted, and I selected the one for organizations. The screen immediately displayed the first five lines of the prompts and showed the length of each field by a series of underscores.

Each time I selected a template or a field, the program would ask some simple questions like the name I wanted to assign to the field, the kind of field (alphabetical, numeric, alphanumeric, etc.), and its length. Since I hadn't read the manual carefully, I started my *field tags* with UPPER case letters.

I later found out that UPPER case letters in fields and templates are distinct no-no's for Perfect Filer. If you use an UPPER case letter in a field tag, the program will be unable to select on that field when you go to produce a report. (I don't know why they didn't write the code to automatically convert UPPER to lower case so I didn't have to spend so much time fixing my fields and templates.)

A particularly handy template is the dtenter tag, which the manual said I should put in the lower right hand corner of the record. When you specify the dtenter field, Perfect Filer automatically puts today's date in that field when you enter a new member (Perfect Filer refers to records as members).

I figured the dtenter template would be helpful in my data base, since we would probably be adding members after sending out the first wave of invitations, and I could prepare subsequent mailings by selecting only the members added since the date of my last mailing. When I had finished, the screen looked like figure 1.

I pressed the ESC key, and sure enough, I was returned to a menu. The ESC key is a very friendly key in Perfect Filer; it returns you to the next higher menu in the program, carefully asking you if you want to abandon any changes made since the last time you saved to disk. I saved my screen design and went on to data entry (pressing ESC until the program returned to CP/M, then running the program by typing *FILER* again).

With the benefit of hindsight, I recommend that while you're still in the CREATE program, obtain a hard-copy printout of the names you assigned to the field tags, *before proceeding further*. Chances are good that, no matter how good a mnemonic you have chosen, you will forget exactly what you called a field, and will have to exit the FILER program and return to CREATE to see what the names were. You may as well print out the list now, and save the trouble later. A sample printout of the member elements and logical field tags is in figure 2.

Data Entry

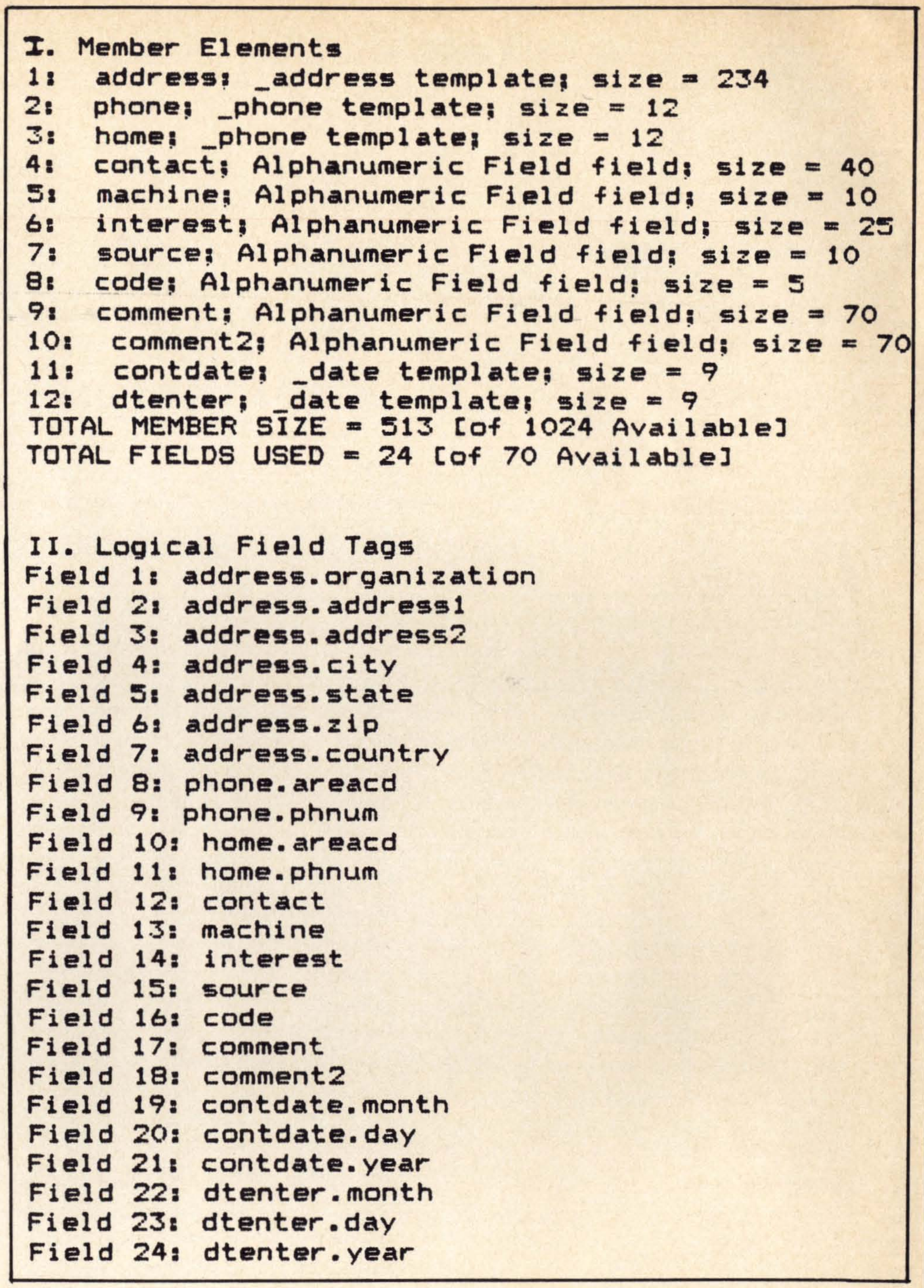
Once the screen was set up, I began entering data. This is one of Perfect Filer's strong points, as far as I'm concerned. Once I learned that the phone template requires an area code followed by a RETURN followed by a seven-digit number with no RETURNs, I had no trouble entering a large number of records. (Too large, perhaps, but more on that later.)

The cursor keys did pretty much what you'd expect them to do, and correcting errors as I went along was easy. If you don't like what you entered in a particular field, press the up arrow until the cursor is at the beginning of the field you want to change. To get to the beginning of the field you're in now, type an up arrow and a down arrow.

Correcting an entry is just a matter of retyping it, pressing the space bar to blank out any remaining extra text. Or you can go back and correct one letter in the field with a strikeover, press RETURN, and the whole field is retained correctly.

When the record looks the way you want, you can enter it in the data base with a Control-*E*. Or you can just type ESC, then follow the friendly menu to [X] Save member data.

Incidentally, my KayPro (a newer model) has a very sensitive keyboard, and I occasionally hit *W* along with *E* I intended to hit for the Control-*E*. That's how I learned how to write the current record to the printer, which is kind of a neat feature — if you're going along and you come across a record you'd like to print out, you can type Control-*W* to print the record. You are returned to your place in the program when the printer is through.



```
I. Member Elements
1:  address: _address template; size = 234
2:  phone; _phone template; size = 12
3:  home; _phone template; size = 12
4:  contact; Alphanumeric Field field; size = 40
5:  machine; Alphanumeric Field field; size = 10
6:  interest; Alphanumeric Field field; size = 25
7:  source; Alphanumeric Field field; size = 10
8:  code; Alphanumeric Field field; size = 5
9:  comment; Alphanumeric Field field; size = 70
10:  comment2; Alphanumeric Field field; size = 70
11:  contdate; _date template; size = 9
12:  dtenter; _date template; size = 9
TOTAL MEMBER SIZE = 513 [of 1024 Available]
TOTAL FIELDS USED = 24 [of 70 Available]


II. Logical Field Tags
Field 1: address.organization
Field 2: address.address1
Field 3: address.address2
Field 4: address.city
Field 5: address.state
Field 6: address.zip
Field 7: address.country
Field 8: phone.areacd
Field 9: phone.phnum
Field 10: home.areacd
Field 11: home.phnum
Field 12: contact
Field 13: machine
Field 14: interest
Field 15: source
Field 16: code
Field 17: comment
Field 18: comment2
Field 19: contdate.month
Field 20: contdate.day
Field 21: contdate.year
Field 22: dtenter.month
Field 23: dtenter.day
Field 24: dtenter.year
```

Figure 2. *Sample Perfect Filer printout of member elements and logical field tags.*

Printing a Report

After I entered many names, I decided to print out a list of all the members entered, in order to find and eliminate duplicates and check potential new entries against those already entered.

Before I printed the report I needed to design a list format so the report would look the way I wanted it: in zip-code order, and with each member on one line along with some determining information, like zip code, name, and city.

After several not-helpful error messages from Perfect Filer I stumbled onto the solution, more or less by accident; I thought it was unfair of the Perfect People to ask us to read the entire Perfect Writer manual in order to use the Perfect Filer program.

I chose my first field as a data field, and specified it as *address.zip* (wasn't it handy to have printed out the list of field tags when I first created the data entry screen?). Then I hit ESC until I was invited to add another field.

I made the second field a string, identified it as a comma followed by a couple of spaces, and hit ESC to return to the menu. Another ESC took me to another field.

I specified a data field again, this time *address.organization*. After pressing ESC, Perfect Filer invited me to do another. I typed another string with a comma followed by a couple of spaces, then *address.city*, a series of ESC keys, and one more string: this one a RETURN.

Notice that, although all these fields are part of the address template, they can also be specified as individual fields. I typed ESC again until I got to the menu, where I could specify sort keys.

It takes two ESCs to get to the next field; if you only type one, you have the menu again for the field you just entered. This can be somewhat confusing; but if you check the upper left-hand corner of the screen, you'll see the number of the field you're currently working on. (If you only type one ESC, you can then choose to specify report parameters like the length of a field, whether it should be centered, left-justified or right-justified, etc.)

Since I wanted the items in zip-code order, I specified *address.zip* as the field for the program to sort on. I returned to the menu and typed an *X* in the choice marked [ ] Save list format. Printing it out proved to be as easy as it looked.

Updating Records

When the list was printed out, I noticed a few typos here and there, and decided to correct them. The menu seemed clear enough — [X] Review/Update Existing Member. And, indeed, the next prompt was easy, too: Enter first four letters of address.organization:.

Many of my organizations began with the same four letters. This was my first introduction to the Control-*O* command (to go on to the next member). I had about twenty names beginning with Appl, and Control-*O*-ing through them was much less fun than entering them.

For me, the member ID feature is one of the worst aspects of Perfect Filer. It could take forever to page through a data base looking for one member starting with *Comm* after you've entered fifty Commodore users' groups.

The problem is even worse if you find you have duplicates you need to delete from the list. Not only do you have to type a Control-*O* after each record you don't want, but the screen comes up with a message that I'm sure is intended to be helpful, but isn't: You will be shown a data display screen—to delete the member displayed type Control-D (the Delete command). Hit any character to continue. After a while, you get in the habit of pressing the spacebar immediately after each Control-*O*, so you don't just sit there watching the screen wait for you to hit any character to continue.

Backups

Each time I finished entering a set of names into the file, I made a copy of my data disk. Backing-up data is absolutely imperative.

You should copy your data disk any time you've made changes you don't want to lose. The most amazing things can happen to the one-and-only-copy of that vital disk that you meant to back up tomorrow!

Fortunately, the KayPro copy utility on the CP/M disk is quick and easy. With the CP/M disk in drive A, press reset and type *COPY* when the A⟩ appears. From the COPY menu, select the copy option by typing a *C*, then replace the CP/M disk in drive A with your original data disk, and place your backup data disk in drive B. Press RETURN, and the program does the rest.

The Impossible Task

The next task I attempted proved impossible. I wanted to print out a list on gummed labels in zip-code order. Sounds simple, right?

Each label has nine lines on it. I followed the instructions for list formatting with continuous form labels, inserting what appeared to be the correct number of carriage returns, then put some labels in the printer, and ran some to see how they looked. There was one line too many on each label, so they soon were out of sync and were printing the last part of one address on the top of a label, and the first part of the next address on the bottom, with the middle of the addresses printed in the space between labels. Well, *that* certainly wouldn't do!

I went back to define list format, and, to get one fewer line, took out a RETURN (carriage return). This time the program printed one too *few* lines, causing the same kind of skewed result as before.

I went back and forth for several hours, first adding a RETURN, then deleting one, first in one field and then another, until I decided to go home and call KayPro in the morning and see what they could do. (I had already called Perfect Software's technical support people, and they had told me my program disk probably had a bad sector or some other similar glitch... Hmph. I wonder how they're going to learn about the bugs in their programs.)

"That intrigues me,“ said the nice lady at KayPro, who earlier had clearly and kindly explained in perfect detail how to remedy my disaster caused by UPPER CASE letters as field tag identifiers. "I don't know what you can do. Why don't you print them out with the send mail option instead of as a list?“

The only problem with that solution was that the output would not be in zip-code order, and since I thought we might use a second-class permit, order seemed important. But I saw no way around it, so I created a print format, using the continuous forms and labels options for nine lines between labels.

So I printed the labels out in the order they'd been entered, and at least got them ready by the time the guidelines came back from the printer.

A "Gotcha“

I found one other major drawback to the program. Unlike some other file management systems I've used, Perfect Filer gives you no indication of how many records you've entered in your data base. I wouldn't mind, except that I accidentally over-filled my disk, with unfortunate results.

It seems that although Perfect Writer will accept as many records as the disk will hold, when it comes time to sort them, you can run into trouble. If you have more than some mysterious number of records (I guess about 550-600, based on what it did to me), it goes through the selection and sorting process fine, until it gets to the *very last step* of the shell sort or the heap sort, whereupon it gives the alarming and not very helpful message, Warm boot (I'm not sure where it uses the shell and where the heap; it bombed out on both of them at one time or another).

Error trapping is not one of Perfect Filer's strong points. I saw that there were over 1,000 records in the file, which seemed like a lot to me, so I figured I must have given it more to sort than it could handle. "I'll divide it in two," I thought.

The Solution: Subsets

Perfect Filer makes it easy to divide the data base, by letting you create what it calls *subsets*. I followed the menu and defined my subsets as those with zip codes less than 50000 and those with zip codes greater than or equal to 50000.

The subset definition option lets you define a "relational operator" by placing the cursor within a field and typing a Control-*R* The screen then displays a menu with the various relational operators (equal, less than, less than or equal, etc.) from which you choose with the familiar *X*. You can define your primary criteria, say, greater than 0, and then define your "limit criteria", say less than 50000.

Following a similar process, I defined the two subsets. The only problem was that I hadn't made them small enough: once again, at the *very end* of the process, the wretched message, Warm boot again. I felt like putting a warm boot through the screen.

I persevered (what else was I going to do?). I needed a list in zip-code order, or I would never be able to tell whether the current magazines' lists of users' groups had any new ones for me or not. So I re-defined the subsets as from 0 to 19999, from 20000 to 39999, etc., and this time it worked.

I printed out my list, and there was no hiatus in the printout to mark all the hassle I'd had to go through: the printing option allows you to specify subset after subset, and crank them right out one after another without skipping a line.

The subset definition (using relational operators) was also useful for printing out a list of the people for whom I had no zip codes. I entered a "0" in the field, pressed Control*R*, and selected less. Then I could print out the name, street address, city and state on one line and have a handy reference to use with the zip code directory.

I used the same technique in another project, this time to list the magazines whose address I needed to look up. I typed an *A* in the city field, pressed Control*R*, selected less, and obtained the subset of magazines with an empty "city" field.

Using Perfect Writer Files

Sometimes, Perfect Filer requires that you put some information in a file using Perfect Writer, which Perfect Filer will then access. The send mail option, for example, calls for format files created by Perfect Writer, and if you're not a careful manual-reader, you may not notice that the Perfect Writer file needs to have the file type ".MSS" after its name.

After several unhelpful error messages from Perfect Filer like "unable to open format file," I stumbled on the solution, more or less by accident; I thought it was unfair of the Perfect People to ask us to read the whole Perfect Writer manual in order to use the Perfect Filer program. Of course that's no problem for all good people who always read all the documentation before touching finger to key. For the rest of us, remember to add ".MSS" to the file name when you're making your format file using Perfect Writer.

Changing a Field

Perfect Filer is very forgiving of errors made in designing the input screen. After entering several dozen members, I decided to add a field to the records. I hadn't included a second comment field, and I figured we might want more than 55 characters of comment. Fixing this was easy, using the third program on the Perfect Filer diskette — MOVE.

Before attempting to run the MOVE program, you must make a copy of your data disk. The MOVE program will destroy the data on your present disk, so it is very important to make a backup. I even recommend making two backups before initiating this procedure. (Much as I dislike reading manuals, I dislike retyping hundreds of data entries even more!)

I followed the simple menu instructions, and the MOVE program worked fine. It re-copied all the data I'd typed before, leaving the new field blank. That was fine for me, since the comments would all be added later, after responses started coming in.

The MOVE program can also be used to delete a field you no longer need; since it only copies data from one field to another field with the same name, if it doesn't find any field with that name, it will go on to the next field.

Overall Feeling

There is no HELP command in Perfect Filer, which is too bad. A good help facility would tell me my disk was too full to try such a sort; or it might suggest how I could get my labels printed in zip-code order. It could even remind me to avoid using UPPER CASE letters in naming my field tags, or point out that my format file lacked the requisite ".MSS" suffix.

However, I enjoyed learning how to use Perfect Filer. Once I was aware of its little traps, I could step carefully around them and still produce the output I required. I made some mistakes and encountered some interesting idiosyncrasies, but creating and maintaining a simple data base was a relatively easy task. In retrospect, I'd even say it turned out to be fun.

Although we have about a dozen data base and file management systems in the office now (we are compiling software reviews), I still use Perfect Filer for simple jobs needing only one file. Perfect Filer is quick and perhaps somewhat dirty, but it works.

FOG

The **FIRST OSBORNE GROUP (FOG)** has been formed as a User's Group for persons using or interested in portable computers and/or the CP/M disk operating system with related software. In September of 1983, the membership topped 5400 with members from around the world.

Computer systems owned or used by members include the Osborne 1 (single and double density), the Osborne Executive 1, all models of the Morrow MicroDecision, the Zorba, all models of the KayPro, several Chameleons, plus many more. Special interest groups organized to augment a network of local group meetings include dBaseII, Ham radio operators, Personal Pearl, and many others.

FOG was started in October of 1981 by a small band of early buyers of the Osborne 1. The primary purpose was to organize a library of public domain software to run on the Osborne.

A newsletter was quickly started to act as a focal point for the group's activities. The large number of excellent contributions to both the library and the newsletter has produced a library of 111 disk (as of August 1, 1983) and nicely typeset (24 ormore pages) newsletter which is published monthly. All back issues of the FOGHORN are available for a nominal fee which included shipping in the U.S.

The FOG library is maintained on the Osborne 1 single density format but separate libraries are being established for the other computer formats. The library disks are carefully screened and divided into category types (utilites, games, applications, and computer languages are the four major categories. Programs which contain run or other errors are put into the hacker section so interested members can fix them and resubmit for inclusion in the correct section. Items which do not fit into one of these categories are in the miscellaneous section. A catalog and descriptions of all the discs is maintained in the library section.

While the meetings are organized on a local basis, many of these local groups are joining the FOG network, thus increasing the sharing of information, tips, problems and so on. Those local groups which opt to formally join the FOG network receive a portion of local member dues to assist with the cost of maintaining a local copy of the disk library.

Dues in FOG are $24.00 per year. This entitles each member to a copy of the FOGHORN each month as well as access to the disk library. Most local group meetings are open to the public without charge although most restrict access to the library to the membership. The FOG library contains only public domain software. Piracy (the copying of proprietary software) is strongly condemned.

Contributions are currently being solicited for a bulletin board system which we hope to have in operation in the fall of 1983. It will accept both 300 and 1200 baud calls.

In the United States, the FOGHORN is normally mailed by non-profit bulk mail. (FOG is a corporation in the state of California and has obtained it non-profit, tax exempt status from both the state and federal governments.) For those members who live out of the country or who prefer first class delivery of their FOGHORN, additional postage must be added to the annual dues. See the chart below for details.

If you are interested in joining a self-help organization to increase you knowledge and the use of your computer, use the applciation below (or a copy of it). Generally, memberships received at the FOG office prior to the 15th of the month are entered in time to receive the next month's FOGHORN.

If you know of a local group which might be interested in joining the FOG network, please send all details (meeting dates and places, officers, and how interested local computer owners can join). We will send you an information packet on becoming an Affiliated Member Organization.

For your records, the address of FOG is P. O. Box 3474, Daly City, CA, 94015-0474. Please allow at least two months for the arrival of your first FOGHORN since bulk mail can take as much as nine weeks. (The post office says that it should only take about three week for non-profit bulk mail but some members on the East Coast have experienced longer delays.) A membership card will be processed within a week of the receipt of your dues.

ADDITIONAL POSTAL CHARGE CHART

Please add the appropriate amount to your dues payment.

Canada & Mexico (First Class Airmail delivery)	ADD...$ 6.00
Members with U.S. addresses who prefer First Class delivery to bulk mail	ADD...$ 6.00
Central & South America, Caribbean, & Europe (Airmail First Class delivery)	ADD...$12.00
Asia, Africa, & Far East (Airmail First Class)	ADD...$15.00
Out of North America preferring surface mail — delivery not guaranteed	ADD...$ 6.00

CUT HERE

NAME: ______ COMPUTER TYPE? ______

COMPANY: ______ MODEM TYPE? ______

ADDRESS: ______ PRINTER TYPE? ______

CITY: ______

STATE: ______ ZIP/MAIL CODE: ______ COUNTRY: ______

HOME & WORK PHONES: ______

MEMBER OF LOCAL GROUP? ______

INTEREST? ______

Send this completed application **AND** your payment to:

FIRST OSBORNE GROUP
P. O. Box 3474
Daly City, CA 94015-0474
United States of America

Circle number 8 on the reader service card.

CP/M Plus: The Inner Workings

How CP/M Plus Controls Extra Memory, and What Hardware Resources Are Required

by Steve Rosenthal

(C) 1983 Steve Rosenthal

CP/M is a registered trademark of Digital Research, Inc. (Pacific Grove, CA).

*The last installment of this CP/M Plus overview (*User's Guide *no. 4) described how the system appears to an everyday user, and some of the differences between CP/M Plus (CP/M 3.0) and prior versions of CP/M. This section continues with a closer look at the inner workings of a few of the new parts of CP/M Plus.*

The goal in designing CP/M Plus was to make the system more powerful and easier to use than earlier versions of CP/M. To make this growth possible, the internal complexity and resources needed by the system have also expanded.

Several of the changes are major structural ones, and affect both programmers and casual users. Others represent internal modifications that most users can ignore. Both users and creators of programs will notice the expanded possibilities of the system.

History of The Memory Crunch

When microprocessors in the Intel 8080 family were first designed, memory was still quite expensive and typical programs were small. Even in software development, where files are typically large, the usual practice was to do the work on larger computers and then download the result to the microcomputer.

Inside the microcomputer chip, register space and room for data paths were both quite limited. For reasons related to both hardware design and software, designers decided to use some small multiple of eight bits for memory registers and data paths.

With the pressure on to keep the memory addressing and pointers section of the chip small, microprocessor designers took a look at typical applications and decided that 16 bits of memory addressing was the right amount. Each additional bit of memory addressing allows a computer to keep track of twice as much space; therefore, 16-bit addressing lets the system address 64K of memory (which is actually 65,535, since a "K" is a "binary thousand" of 1,024).

Designers used 16-bit addressing for the 8080 and for all the other popular eight-bit computer chips (including the Zilog Z80, the Motorola 6800, and the 6502 used in the Apple, Commodore and Atari). The designers of computers using these chips also accepted this view, and a 64K system was the maximum amount most designs allowed (although it was several years before memory prices declined to the point where many people could afford to buy that much memory).

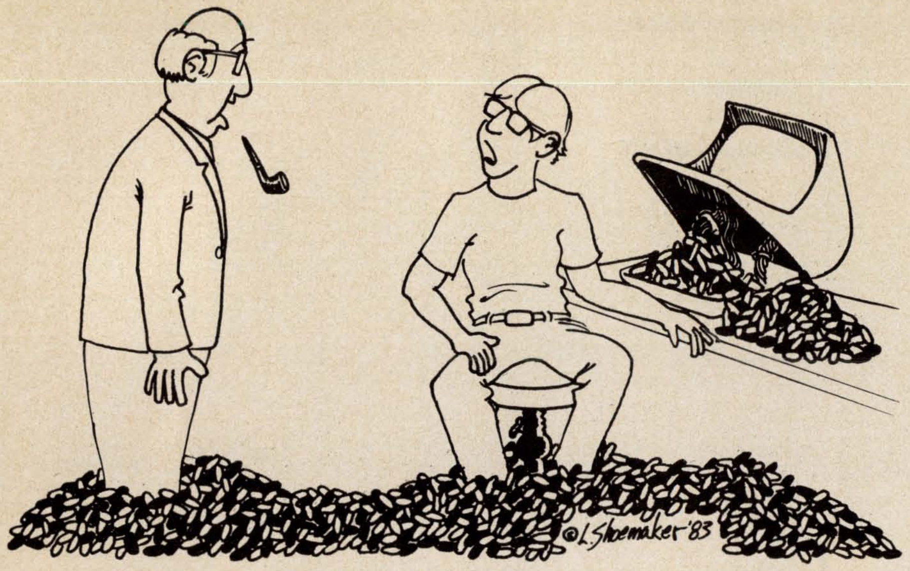
The software for the computers using these chips followed suit. With 64K the maximum system, programs that needed access to large amounts of storage had to be written as chains or overlays or broken up into separate units. The operating systems that were developed, including CP/M, followed this design, and kept all of their workings within 64K by swapping certain parts in and out from disk as needed.

Now that memory is less expensive, and users want more powerful programs and more space for data, how can designers satisfy those wishes by adding to existing hardware and using existing software.

One method that has been widely adopted is to put more than 64K of memory in the computer, wired so that only 64K is connected to the microprocessor at any given time. The most common way of doing this is called *bank switching.*

Bank Switching

With a bank-switched system, memory is organized into two or more blocks, with each block limited to a maximum of 64K bytes. To access different blocks, the system writes to a



Well YOU told me to check every bit of the program!

special memory or output address which causes the hardware to "de-select" one bank of memory and select instead another. Many systems have only two banks, but some designs offer as many as 256. The banked version of CP/M Plus supports up to 16.

A diagram of a typical banked CP/M plus system is shown in figure 1. It consists of three banks: one mostly for system use, one mostly for user programs, and a third for additional buffering.

With banked CP/M Plus, the hardware does not switch entire banks of 64K in and out, because CP/M needs to keep a great deal of status information in memory, and it requires several thousand bytes of code. If entire banks were switched, much of this information would have to be duplicated, since the system needs to be able to access the current status information regardless of which bank is active.

The hardware switches in *most* of memory — CP/M Plus recommends 48K to 60K — and leaves the top section of memory permanently in the address space. This allows the status information and some of the system code to appear in all address spaces, without the overhead of making duplicate copies and keeping them updated.

In CP/M Plus, bank 1 (plus whatever part of the common memory that is not needed for the system) is allocated to user programs. When this bank is selected (a state which the CP/M manuals refer to as "in context"), the memory space looks just like an unbanked system with a large user area for programs.

At the bottom of the bank, the page 0 area (the memory locations 0000 through 00FF hex) contains the pointers and status information as in older versions of CP/M. Also, at location 0 is the jump to the warm boot entry where programs return control to CP/M when they finish running.

Location 5 contains a jump to the BDOS (the core part of CP/M called the Basic Disk Operating System), so programs written for earlier versions of CP/M can obtain services from the system by calling location 5 with registers set to the appropriate parameters. Various other parameters fill the area up to 80 hex, where the default buffer extends up through 00FF hex.

The TPA (the transient program area where all programs except the system are loaded) extends from 100 hex up through the start of the operating system, just as in unbanked systems. Note that the TPA may extend into the common memory area, because this part of the common area is left unchanged by the system when other banks are active.

The System Area

One big advantage of a banked system is that only a small part of the operating system need be in the common area, leaving room for a bigger TPA. In the banked version of CP/M Plus, only 1.5K of code and date is placed in the common area. The rest of the resident operating system code is placed in bank 0.

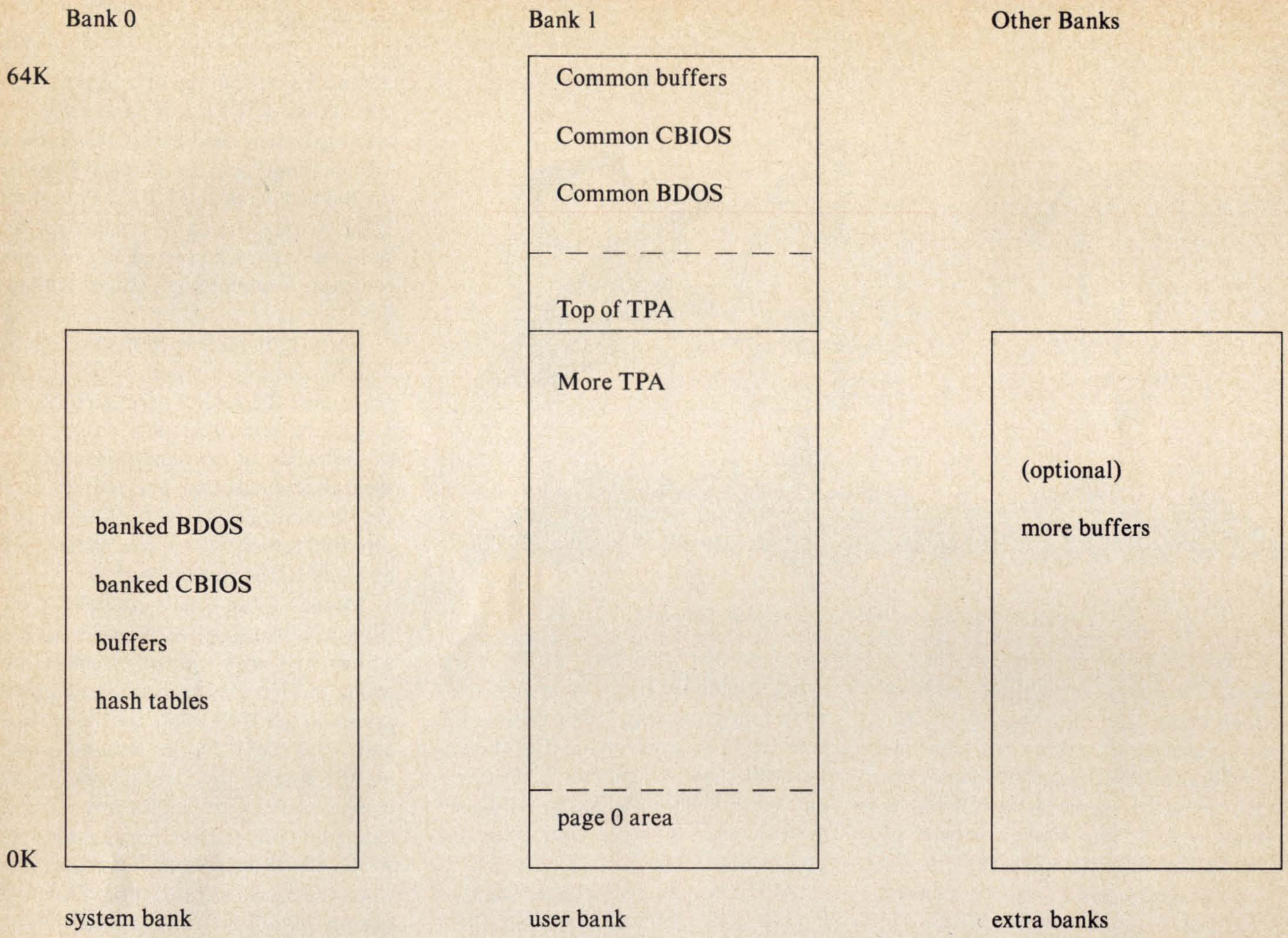
Entry to bank 0 is from the system area in common memory. After a program calls the system through location 5 to ask for some operation, a small section of code in the common memory area checks whether the request is a simple one that can be handled in the common region. If not, the system sets up a call to the system bank.

Many system calls require data from the transient area, such as file names and disk control blocks (the copies of directory entries used by programs to tell the system where to read and write files). Before actually making the switch, the system code in the common area first copies any needed data into common memory or bank 0, so it will be accessible for processing when the banks are switched. Then the system switches banks, and jumps down to the base of the system area in bank 0.

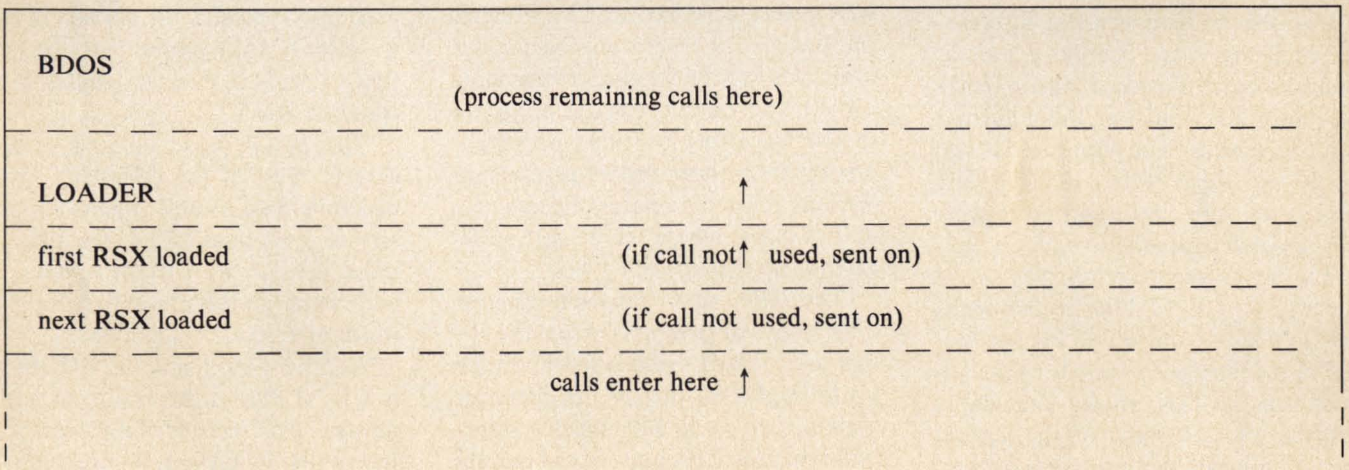
Because the size of the system code in bank 0 does not subtract from the memory space available for user programs, the banked CP/M Plus BDOS can be larger than allowed in previous versions. Typically, the portion in bank 0 is close to 11 thousand bytes (in addition to the 1.5 thousand bytes in the common area).

The BDOS code in bank 0 takes care of most of the operating system disk functions, as well as assorted other service functions. The routines are quite similar to CP/M version 2.2, with some added processing for passwords and access restrictions.

Another addition to the BDOS code in CP/M Plus is the ability to handle physical disk records of different sizes. Internally in CP/M, all records are considered to be 128 bytes long. But most disk systems now use bigger blocks, with sizes of 256 bytes and 1,024 bytes most common. In CP/M


Bank 0
Bank 1
Other Banks
64K
Common buffers
Common CBIOS
Common BDOS
Top of TPA
More TPA
page 0 area
banked BDOS
banked CBIOS
buffers
hash tables
(optional)
more buffers
0K
system bank
user bank
extra banks

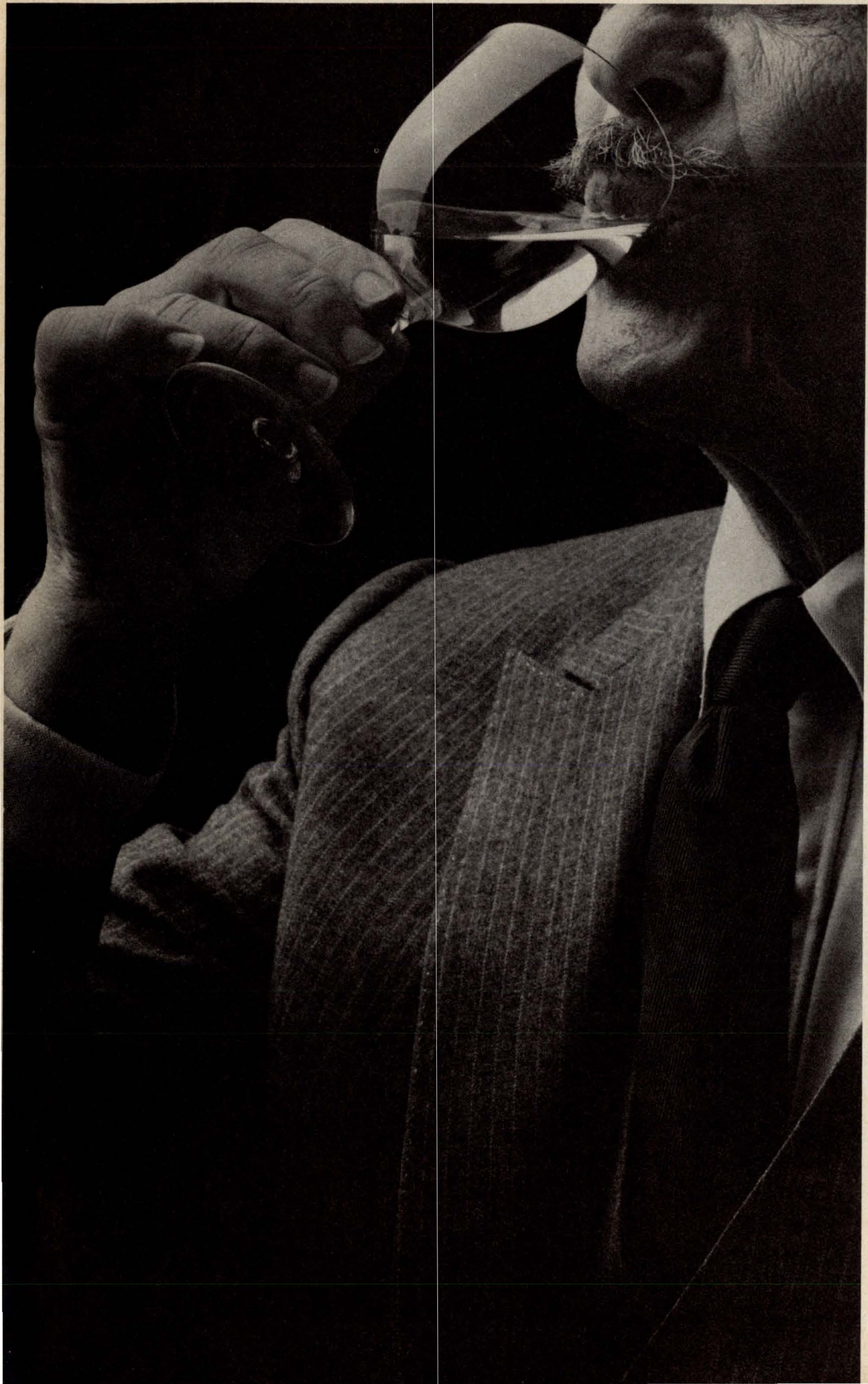
By dividing memory into banks, CP/M Plus allows using both large programs and lengthier system code.


BDOS
(process remaining calls here)
LOADER
first RSX loaded
(if call not used, sent on)
next RSX loaded
(if call not used, sent on)
calls enter here

Each active Resident System Extension in turn has a chance to intercept the normal BDOS call.

Figure 1 *A typical banked CP/M Plus system consists of one bank for system use, one for user programs, and one for additional buffering.*

THE SUPERB QUALITY OF THIS PRIZE CHARDONNAY IS ANOTHER MILESTONE® ACHIEVEMENT.



While the grapes struggled to grow and mature, the vintners were struggling with a problem of their own.

The problem was *When?*

When to pick. When to crush. When to test. When to taste. When to bottle. When to age. When to ship.

And when to wait.

And each variable affected the cost and release date of what promised to be their prize Chardonnay.

Fortunately, they used Milestone Project Management Software.

Milestone told them when. And helped the vintners manage the project right down to the day when they savored their first glass.

You can savor the rewards of Milestone whatever field you're in—banking, building, retailing, manufacturing.

Milestone plans the lifetime of your project, and streamlines its schedule, by analyzing its "critical path." Milestone finds which activities are crucial, prepares cost estimates, keeps track of progress, makes trade-offs, and recomputes the project schedule when anything changes.

Milestone's price? Just $295.

About what you'd expect to pay for a few bottles of Chardonnay.

Prize Chardonnay, that is.

For more information about Milestone, call

TOLL-FREE 800-826-2222.

Or write to:

Digital Marketing Corporation
2363 Boulevard Circle,
Walnut Creek, CA 94595
(415) 947-1000/Telex 17-1852

Milestone requires 56K RAM and CP/M,® CP/M-86® or UCSD-P™ system. CP/M-86 version requires 128K RAM.

DIGITAL MARKETING

Circle 6 on the reader service card.

version 2.2, assembly and disassembly of these larger physical records had to be done in the CBIOS section written for each specific system (although Digital Research furnished a sample routine and information about how to write a custom version). In CP/M Plus, this can be done internally by the BDOS.

The Banked CBIOS

Just as in previous versions of CP/M, every CP/M Plus system consists of the standard parts provided by Digital Research, plus the Customized Basic Input/Output System (the CBIOS) written for each type of machine. These routines adapt the standard operations of CP/M to the workings of the circuitry and configuration of a particular machine.

In CP/M Plus, the CBIOS may also be banked, with a small part placed in bank 1 with the user area, and the rest put in bank 0 with the system area. This also contributes to a bigger area for user programs, and a more complex CBIOS.

You can split the CBIOS into two parts by assembling the code with the provided RMAC relocatable macro assembler, and then using the GENCPM utility to allocate each section correctly. If you have other banks available, you can add the needed bank-switching routines and put your CBIOS in any of banks 2 through 15. This allows the construction of a CBIOS up to dozens or even hundreds of kilobytes long.

In theory, user programs can directly make use of bank switching. Although there is no BDOS call for bank switching, the BDOS does allow calls to the BIOS section, and one of the new BIOS calls is the bank switch request. However, the BIOS function does not return any status information about whether the requested bank is available, nor about how large a space is available in any bank. This limits bank-switched programs to ones that are configured for a particular hardware environment.

Directory Hashing

Searching for a directory entry is one of the slower operations of prior versions of CP/M. Entries are fields on the disk that tell CP/M where to find files and which parts of a disk have already been used. CP/M must frequently read them in from the disk, and they must be updated and replaced on disk whenever permanent changes are made in a file.

In the initial design of CP/M, the decision was made to keep most of the information about file location on the disk, rather than saving the information in memory until the end of the session. That way, if a later program crashed, previous file information would still be accurate. To conserve space, entries were not duplicated in memory, nor were they re-ordered from the way they accumulated on the disk due to additions and deletions.

To make use of an existing file, CP/M needs a copy of the directory entry for that file, and the entry is one of a list kept in random order on the disk. To find the correct entry, prior versions simply started at the top of the list, and read in directory sectors until one was found with the needed information. As a result, on the average the system had to read half the directory sectors every time it wanted to find a file.

CP/M Plus provides a faster way of accessing directory entries. As each directory entry is written to the disk, a sixteen bit number representing a condensation of the entry is written into a special table, along with a pointer to where the complete directory record can be found. The method used to condense the complete file name to just 16 bits is called *hashing*, and the table is called a *hash table.*

When CP/M Plus needs a directory entry, it computes a hash value for the needed entry, checks the table, and if it finds a matching entry goes directly to the sector specified by the hash pointer. In the rare cases where two different file names condense to the same hash value, the system discovers that the name found is not the one being searched for, and reads another entry from the table.

Reading directly into the directory sector saves more time than the simple ratio of a long search to a single (or occasionally dual) sector read would indicate. Because CP/M Plus also saves recently-accessed sectors in memory (as explained below), in a large number of cases the entry is still in memory and no actual physical read of the disk is needed.

Data and Directory Buffering

The other factor that improves the performance of CP/M Plus is its support of data and directory buffering. When the system is set up for a particular machine, any extra memory can be assigned for use as directory or data storage.

Whenever a directory or data sector is read from the disk, it is also copied into the appropriate buffer. If no buffer is free, the system examines the least-recently used buffer (LRU). If the LRU buffer has been altered and belongs back on the disk, it is written out; if the information is unchanged or has only been read, it is simply discarded.

Once a sector is in the buffer, access to it is much faster (a computer can read and write to memory typically 50 to 50,000 times as fast as to a disk). The information can be read, re-read, updated, or replaced without waiting for the disk.

When an internal request from CP/M or a direct request from a program asks that a sector be written out to the disk, the system decides whether to write it to the buffer or directly to the disk. Writing to the buffer is much faster, and frequently, information which is written must be changed again or re-read. On the other hand, information in memory disappears if the power fails, and may disappear if the program crashes.

Consequently, CP/M Plus keeps track of two different types of requests to write to the disk. Whenever a program or the system writes to the directory, that information is considered essential and goes directly to the disk. When the information is intended for the rest of the disk, the system writes it to the buffer area unless the program requests a special write operation.

Some implementations of CP/M Plus automatically keep track of how long any information has been written to a buffer rather than to the disk. At the end of a certain interval (typically a few seconds), the buffers are dumped to the disk even if the space is not needed. This allows for fast access if a program does frequent writes and reads, but keeps much of the data security of direct disk operations.

Resident System Extensions (RSXs)

Although it is a theoretical goal of any operating system to handle all reasonable requests that might be made by a program or user, no practical system can possibly include code for every case. Furthermore, even when the code is there, there may be times when you want to change the way the system handles certain requests.

CP/M Plus handles both these situations by providing for RSXs (resident system extensions). These are special routines which can be created and loaded into the system, where they act like a part of the operating system. They can add to and change the operation of system functions because they get a chance to intercept requests to the system before the normal BDOS section of CP/M.

In operation, RSXs sit in memory below the operating system (the pointer to system entry at location 5 is adjusted to send system calls to the RSX). The RSX examines the each call, and passes through ones which it was not designed to handle (it can also pass through ones it works on for further standard processing).

You can use more than one RSX, with each new one stacked below current RSXs in memory (see figure 2). The RSX added last gets first chance to intercept any system calls, because the pointer is adjusted to the the bottom of the descending series. If the call is not claimed by the first RSX, it goes up the line, and if not claimed by any other, it goes to the BDOS.

RSXs are called into memory by special coding that can be placed at the start of any program. For cases where the RSX is intended for general use rather than with a particular program, the RSX can be attached to a program which consists only of this special code. The actual reading of the RSX is done by the Loader, which is a special RSX which sits below the normal BDOS.

Each RSX contains a flag that tells the BDOS whether to erase the RSX after the current program, or to leave it in memory. If the RSX is set to stay in memory, it remains active until the next cold boot (which happens at reset or powering up the system).

System RSX Usage

CP/M Plus uses RSXs for the GET, PUT and SAVE commands. GET tells the system to take input from a file rather than from the console, and PUT sends output to a file rather than to the console. SAVE tells the system to place the current contents of an area of memory in a file.

The operation of the GET utility is typical of how RSXs are used. GET consists of a program part, a special header to note that it has an RSX, and the RSX itself. When you call GET, the program part is placed at 100H in the Transient Program Area as with any other program, while the RSX is placed at the top of the TPA just below the BDOS.

When the system turns over control to the GET program part, the GET program requests the system to find and open the file you specified to get further input from. If the file is found, the program then prepares the RSX portion to intercept subsequent requests for keyboard input.

As the GET RSX responds to console input requests, it also checks for end of file, and for the command *GET CONSOLE* (or in long form *GET CONSOLE INPUT FROM CONSOLE*). If it reaches the end of the file or finds the command string, the GET RSX stops intercepting console character requests. It also sets a flag saying that the loader can remove the GET RSX during the next return to the system (warm start).

Banked memory and data buffering are important changes, but they don't necessarily force you to change what you do as a user or an application programmer. There are, however, a number of changes in CP/M Plus that can affect the way you write or use programs. We'll cover more of these in the next issue.

Attention CB80 & CB86 users:

Here's a field editing and screen management utility that lets you create interactive software in a fraction of the time for a fraction of the cost!

- The Interactive Program Aid (IPA-1) for compiled CBASIC under either CP/M-80 or CP/M-86
- Source and detailed manual for modifying to your special needs
- Quickly and easily displays data fields and literal strings
- Edits for data type (alphanumeric, numeric and numeric strings) minimum, maximum lengths, dates and Y/N
- Allows editing while data is being entered
- Fields and literal strings displayed in normal or high-lighted intensity
- Cursor control features include left-right movement, delete, insert and skip
- Field termination by carriage return or programmer defined function keys
- Virtually terminal independent...works on almost any terminal with addressable cursor

$30.00 plus $5.00 mailing and handling

(IPA is furnished on 8" SSSD, inquire about availability for other media)

Compare to programs costing ten times more!
Put this creative computing innovation at your fingertips...fill out and mail this coupon with remittance today.

COMPUTER THAUMATURGY

Problem Solvers and Consultants to Computer Users

Suite 700 Miami Valley Tower
40 W. 4th Street • Dayton, Ohio 45402

☐ Please send the IPA-1 user's guide (enclose $5.00 to cover costs and handling and mailing)
☐ Please send the IPA-1 Programming Utility (Postage and Handling included)
Enclosed is my check for $35.00
Ohio residents add 6% sales tax

name ______________________
company ______________________
address ______________________
city ______________________
state ______________ zip __________

CP/M, CP/M-86, CBASIC, CB80 and CB86 are registered trademarks of Digital Research.

Circle 3 on the reader service card.

Hard Disks For Small Computers

Hard Disk Technology Unveiled and Explained

by Steve Rosenthal

© 1983 Steve Rosenthal

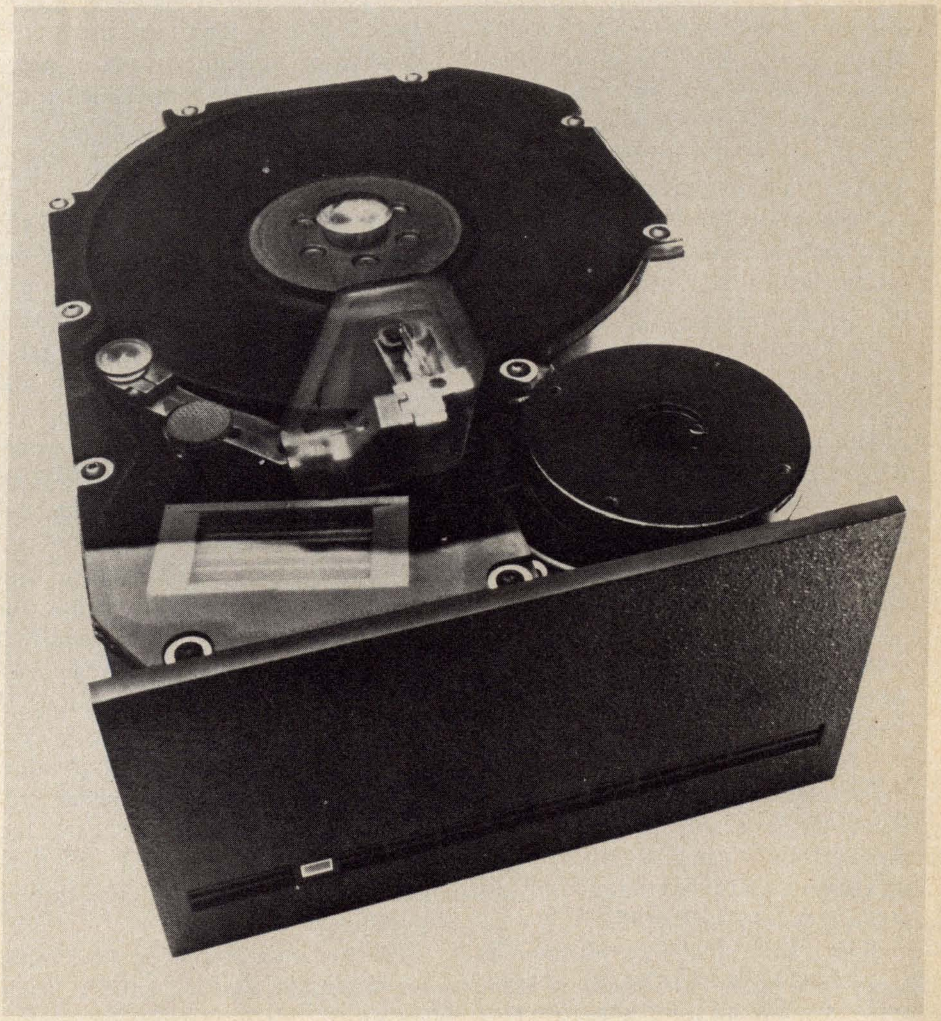
More data accessible at higher speeds — that's the reason an increasing number of small computer users are adding hard disk drives to their systems. The higher capacities and faster transfer rates of hard disks make it possible to use larger files, make programs more interactive, and to switch quickly from one task to another.

Hard disks are increasingly affordable, too. Although not dropping in price as fast as memory or game computers, manufacturers of hard disk drives have managed to make their products smaller in size, higher in capacity and more trouble-free, while greatly reducing the price.

The technology of hard disks is still changing rapidly, so it is difficult to include every feature and characteristics of every model. However, the basic principles remain the same, even as the precise level of performance continues to advance.

The Construction of a Hard Disk

Hard disks, like floppy disks, store data as magnetic patterns on a disk with a thin metal-based coating. In the case of the hard disk, the coating rides on a rigid aluminum platter, which is mounted inside a cartridge or permanently built into the drive itself.



Computer Memories (Chatsworth, CA) 5.25-inch Winchester 40 megabyte hard disk.

Both top and bottom surfaces of each platter are coated, and a read/write head is provided for each surface. In order to increase drive capacity, most companies offer models with more than one platter mounted on the same spindle (the spindle is the center shaft on which the platter turns).

Although larger systems often use hard disks which are eight or 14 inches in diameter, almost all designs for small computers are 5.25 inches or smaller. Both the eight inch and 5.25-inch models are made to fit the space of a similar size floppy drive, thus allowing systems to be designed for a flexible mix of hard and floppy disks.

In fact, eight inch models may be a rapidly dying breed. The large personal computer market directs more development toward the smaller 5.25-inch models for mass-market applications, while the high-performance market for larger disks is going to the 14-inch models.

Winchester Disks

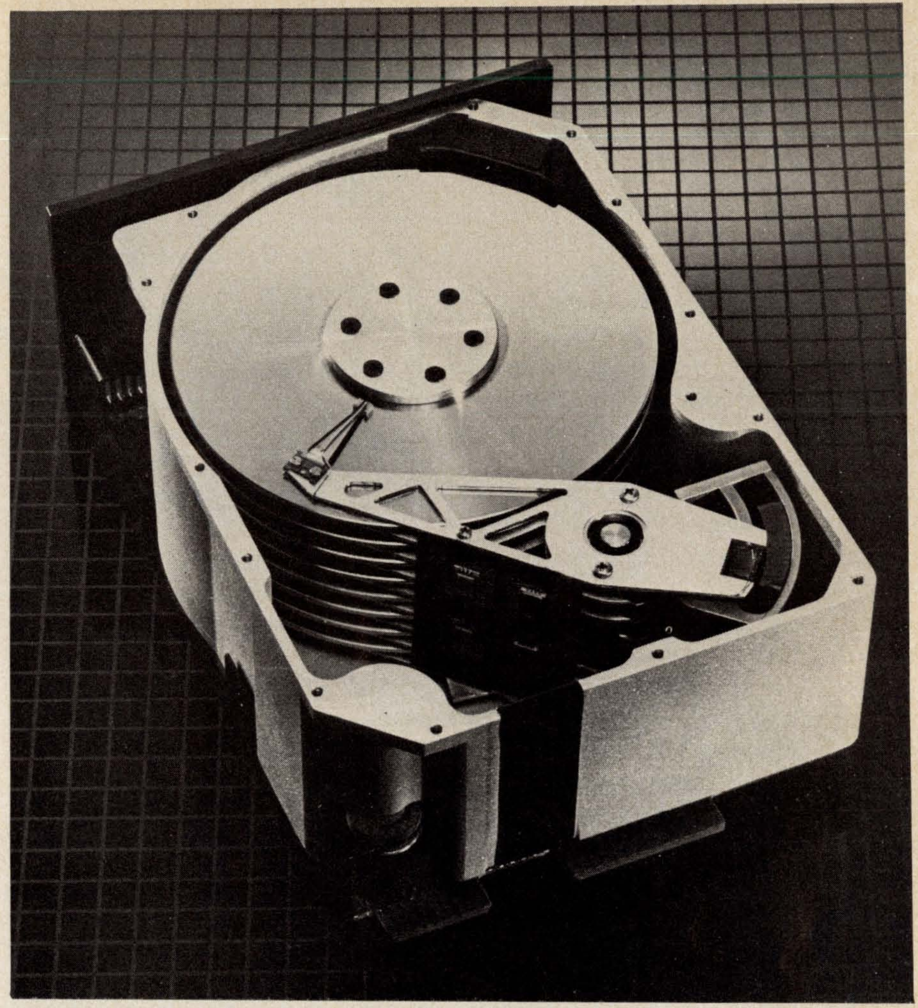
Hard disks are most often classified by the type of technology used in their construction. The term used for each class is usually the model name of the first commercially successful disk of that type. The most common class used with small computers is the "Winchester," named after a drive originally produced by IBM in 1973.

Winchester disks use read/write heads which literally fly over the disk surface from the pressure exerted by the air rotating along with the platters. Since the head does not touch the surface, the platter can spin more rapidly, allowing data to be read and written at higher rates than with a floppy disk.

A high rotation rate is also necessary to maintain the air cushion that keeps the head from touching the disk surface. A typical rate for a small Winchester is 3600 rpm, ten times the standard 360 rpm used for floppies.

Winchester disks also use smaller areas to store bits floppies use. To keep these bit cells small, the coating surface is made quite thin, and the head is made to ride very close to the surface.

In fact, the heads must stay so close to the surface that many of the normal dust and smoke particles in the air will not fit between disk and head. If a particle were to become lodged in the gap, it might tear the disk surface, displace the head, or even twist the head down into the rotating disk. Therefore Winchester drives enclose the disks and read/write heads in a special chamber, into which only highly filtered air is admitted.



Maxtor Corp. (Santa Clara, CA) 5.25-inch Whitney 140 megabyte hard disk.

Whitney Disks

Another major hard-disk technology is called "Whitney," pioneered by IBM in 1979. Whitney disks are an evolution of Winchester technology, offering higher bit densities.

The main distinction is that Whitney disks use a smaller, lighter read/write head than those used by Winchesters. This allows a stiffer, smaller mounting, which in turn allows the head to ride closer to the disk surface.

To make maximum use of this closer head spacing and smaller bit cell, the surfaces of most Whitney-style drives are not made of the familiar iron-oxide coating used on floppies and magnetic tape. Instead, most use a cobalt-iron alloy or other hard magnetic material. This allows the magnetic patterns to be written closer together, and provides some measure of protection if the head should lightly graze the disk surface.

Other Types of Disks

Winchester and Whitney disks both require a closed air chamber containing the heads and the platters (often referred to as the HDA or head-disk assembly). However, there is a decided advantage to making the disk platter removable from the drive, to use multiple platters and make backup copies.

So in the last few years, disk designers have been proposing a number of further variations on the basic Winchester design.

One method is to keep the head and disk assembly together, and make that whole unit removable from the drive case. Although done on some types of large disk drives, the cost of the extra heads and the complex mechanical interface needed has so far kept this from being a popular solution for small computer disk drives.

Another method is to make the drives more like big computer disk-pack drives. Using this technology, the disks are mounted in removable cartridges, and the heads are made to swing or step back completely off the platter for disk removal and insertion. To keep the air chamber free from dangerous dust and smoke, the drive goes through a purge cycle of several dozen seconds after each disk change.

One of the most popular solutions so far is to abandon the hope of keeping all particulates out of the chamber, and make the platter and head capable of withstanding the inevitable collisions. By using a hard alloy plated on the platters and a special head, it is possible to build a drive that just rides over most dust particles.

Actuators

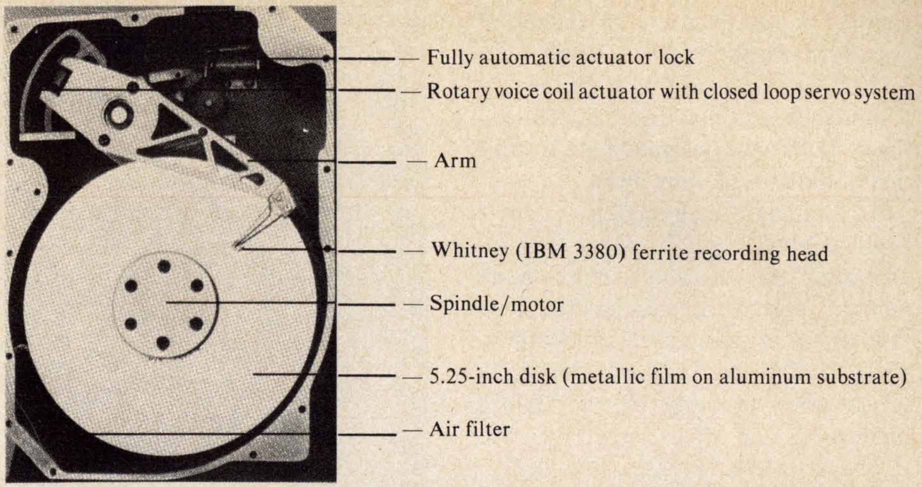
As on a floppy, data on a hard disk is written in concentric tracks on each available surface. To reach each track as requested, the head for that surface must be moved in and out to the proper distance from the edges. Moving the heads can be done in a number of ways.

Larger, high performance hard disks almost all use voice-coil positioning. This method works like a loudspeaker, with the drive electronics energizing a coil which thrusts the actuator forward and backward to the desired spot. Voice-coil actuators are relatively fast, can be electronically varied for different track-to-track distances, and can be hooked to electronics to compensate for slight variations between different platters.

Unfortunately, voice-coil positioners and their control electronics are also expensive, and more difficult to manufacture to the required tolerances in high volumes. For several years, the added cost for a voice-coil positioner was a principal reason why hard disks were economically unattractive for manufacturers and users of small computers.

This outlook was shattered by the introduction of the SA1000, from Shugart Associates. This eight-inch hard disk was a deliberate blend of hard disk and floppy technology, using the sealed HDA of the hard disk and the head positioning mechanism of a floppy.

photo courtesy Maxtor Corp.


— Fully automatic actuator lock
— Rotary voice coil actuator with closed loop servo system
— Arm
— Whitney (IBM 3380) ferrite recording head
— Spindle/motor
— 5.25-inch disk (metallic film on aluminum substrate)
— Air filter

Although not as fast as voice-coils, floppy disk mechanisms were adapted for hard disks to save an appreciable amount of the manufacturing cost of the drive. The positioning mechanism was powered by a stepper motor (which moves in fixed increments), and the head moved in and out on a steel band. The original SA1000 stored five megabytes, but later models had larger capacities.

Since that time, the cumulative experience gained by various manufacturers and the increased sales volumes brought about by a reduction in prices has led to a variety of positioners for small hard disks. High capacity models often use voice-coils, now that large-scale production has brought their cost to more reasonable levels. Many models use a stepper and band, while others use a high performance version of the leadscrew (which moves the head assembly in and out by twisting a threaded shaft).

High Capacity Drives

Using standard stepper motors, iron-oxide coatings and regular read/write heads, the capacity of a hard disk is limited to a few megabytes per platter. Since only a few platters can be squeezed into the equivalent height of a floppy, capacity for an ordinary 5.25-inch inch hard disk is limited to a few tens of megabytes.

In order to meet market demand for still higher capacities, manufacturers have turned to a number of methods. Principal among these are increasing the number of tracks per disk by better positioning, increasing the number of bits per track by changes in coatings, and increasing the number of platters per spindle by re-designing the geometry.

To put more tracks on the disk, most high-capacity models use some kind of *closed-loop* positioning system. On these systems, position information is permanently written into the disk by the manufacturer on either special servo tracks or servo sectors. After the drive electronics move the head to what is expected to be the proper position, the servo information is read and the position adjusted for exact match. This method has allowed some firms to pack on the order of 1,000 tracks per inch.

To put more bits in each track, some manufacturers are opting for plated coatings and thin-film heads (which are made by a process similar to making integrated circuits). Because this method is quite expensive, other companies have instead opted to try to fine-tune the oxide coating and standard head, and couple them with improved electronics. With both methods, information density can also be increased by switching from the MFM encoding used to represent data on floppy disks to the more complex codes used on bigger disks, such as RLL.

To get more platters per spindle, several companies have switched to Whitney technology. As mentioned above, Whitney heads are smaller and stiffer, so they need less room. Still more space has been found by moving the drive motor into the center of the spindle, and the drive electronics in the corners of the case.

Controllers and Interfaces

The controller needed to run a hard disk is significantly more complex than one for a floppy, and only recently have hard-disk controller chips started to become available. As a result, hooking up a hard disk has been more difficult than adding a floppy.

Some manufacturers of small computers have designed dedicated controllers for their systems, and settled for selling fewer systems at higher costs. But since most of the controller circuitry is the same for different models of computers, other companies have opted for supplying a standard controller board, and then a separate *host adapter* for different types of computers.

The most common interface between controller and disk is an unofficial one modeled after the Seagate Technology ST506 disk (the first mass-market 5.25-inch inch disk). This interface includes one 34 pin connector and another with 20 pins, plus a 4 pin power connector. The 34 pin connector carries control and status information in a form much like floppy disks, while the 20-pin connector carries the actual data.

The controller to computer interface generally follows the SASI (Shugart Associates System Interface) model originated by Shugart for their small hard disk drives. This interface has eight data lines, and another eight control and command lines.

Since these interfaces are not standardized, not all controllers work with all adapters or all disks. And as the performance of small disks improves, many in the industry want to see new more powerful interfaces.

At the disk end, a proposal is circulating for an Enhanced Small Disk Interface (ESDI). Although closely modeled on the ST506 interface, this proposed interface would double the data transfer rate from the current ST506 rate of five to ten megabytes per second.

Forming a link between host and controller, the proposed Small Computer System Interface (an updated and standardized version of the SASI interface) would share a common command structure between hard disks, floppy disks, and tape storage. A more complex ISI (Intelligent System Interface) proposal is also under consideration. The *SMD* interface, long used for larger mainframe and minicomputer systems, may continue to be offered on 14-inch hard disks, but it won't be a major factor for personal computers.

Backing Up Hard Disks

One problem that has slowed the use of hard disks for small computers has been the lack of a good means of making backup copies of the information on the disk. The smallest hard disk (with a capacity of five megabytes) stores much more than a 5.25-inch floppy (typically 320K for double-sided double density floppies, equal to 1/15th of a five megabyte disk). Making a complete copy of the information on the disk requires shuffling a large number of disks (*the infamous floppy shuffle — ed.)*

Of course, one way to back up a hard disk is with another hard disk. Most systems, however, don't have space for two hard disks, and you are limited to one backup. The cost of a second drive, even without the controller, also makes this unattractive.

A better way would be with a removable disk. Although these models are still more expensive and less reliable than fixed disks, a growing number of users are opting for this method. Unfortunately, track density must be limited on removable hard disks to allow for the differences between cartridges and differential temperature and humidity effects. As a result, removable hard disks that sell for less than a few thousand dollars per drive are still limited to on the order of a dozen megabytes per cartridge.

Another popular method is to make the backups on cartridge tapes. These data cartridges, which are distantly related to eight-track tapes, now come in versions which store several dozen me-

P&T CP/M®2 is GROWING

TRS-80 MODEL II
$185
Still the best CP/M for the Mod II with features like 596 Kb per diskette, type ahead, full serial port support, and more.

TRS-80 MODEL 12/16
$220
Includes full support for thinline drives; gives 1.2 Mb per diskette for the Mod 16 (Z-80 mode), Mod 12 and Mod II's with double sided drives.

RADIO SHACK HARD DISK
$250
Includes all the features of P&T CP/M 2 plus support for the 8.7 and 12 Mbyte hard disk drives.

CAMEO HARD DISK
Support for the standard Cameo hard disk system ($250) or the multiplexer (for multiple computers) system $400.

CORVUS HARD DISK
$250
Support for a 5, 10, or 20 Mb Corvus hard disk system.

Start with a Model II floppy system and grow into a hard disk. Since all P&T CP/M 2 systems are fully compatible, you will have no conversion worries.

Special note: P&T hard disk systems allow you the user to configure logical drive assignments to your specifications. Write for more details.

Prepaid VISA, M/C, or COD orders accepted. All prices FOB Goleta and subject to change.

CP/M is a registered trademark of Digital Research. TRS-80 is a trademark of Tandy Corp.

PICKLES & TROUT
P.O. BOX 1206
GOLETA, CA 93116
(805) 685-4641

PICKLES & TROUT®

Circle 13 on the reader service card.

gabytes, and take only a few minutes to back up an entire disk.

Cartridge tapes used to back up hard disks normally operate in the "streaming" mode. Instead of starting and stopping for each block of data, the tape is stored and data is written continuously until the end of the transfer.

While this makes it impossible to locate and update data in the middle of the cartridge, much more data can be put on each tape since no space is taken up with the normal blocking and positioning information. As long as the tape is going to be used merely to make an exact image of what is on the disk and then later replay this image directly back to the disk, this lack of internal structure does not present any problem.

One final method, offered by Corvus, has been to copy the disk data to videotape. Because there is a large market for videotape recorders, prices per unit of storage are much cheaper than on any computer-oriented product. However, video-recording technology accepts a high number of small errors in recording and playback, so complex error detection and correction circuitry is required to use it for computer data storage.

Disk Specifications

Hard disks are generally characterized by their storage capacity. In complete systems, the number given is usually the *formatted capacity*, which is the amount of actual data that can be put on the disk in addition to the location and timing information needed by the disk controller. When speaking of the drives themselves, either the formatted or unformatted capacity may be given — formatted capacity is generally from one-tenth to one-half less than unformatted capacity.

Disk rotation speed is almost always 3600 RPM for small hard disks. This speed also determines rotational latency (the time it takes for data to come around to the position where it can be read or written). For a drive with steppers, head positioning time is generally on the order of a few milliseconds or less per track, but many drives use a faster rate for track seeks of more than a few tracks. Voice-coil drives generally claim a step rate about 50 percent faster, but it depends on the distance travelled.

PART NO. 300671
CERTIFIED HIGH DENSITY CARTRIDGE

DEI (San Diego, CA) high-density mag tape cartridge operates in streaming or stop/start mode.

As with floppies, disk error figures are generally specified as the maximum frequency of each of three types of errors: seek errors, soft errors, and hard errors.

Seek errors refer to times where the drive does not position the head to the proper track on command (drive manufacturers claim they occur no more than one in a million moves). The mispositioning can be detected by the controller or software, and by stepping out to track 0 and then back in, the error can be rectified.

Soft errors, where an initial read operation provides incorrect information but retries obtain the correct data, typically occur less than one in 10 billion sectors read.

Hard errors, where a reasonable number of retries do not result in valid data, are typically specified to be about 100 times more infrequent than soft errors.

It is not uncommon to have a limited number of spots on a hard disk which are permanently defective. Because of this, almost all drives include room for a few spare tracks, which are used instead of the defective ones. More sophisticated controllers automatically substitute one of these reserved tracks for the bad one, while in other systems this must be done in the software of the host computer. However, some powerful operating systems (such as several Unix derivatives) work better if the controller does not automatically do the substitution — the operating system itself contains routines for optimizing the placement of information on the disk and skipping over bad sectors.

Hard Disks and CP/M

Supporting hard disks was one of the considerations in revising CP/M from version 1.4 to versions 2.0 and 2.2. By taking the parameters that specified disk capacity out of the core of the disk operating section (the BDOS) and putting them in the (BIOS) interface section, these new versions let CP/M make use of the larger capacity of hard disks, and to skip checking for a change of disks on non-removable models.

Unfortunately, CP/M 2.2 only allowed for disk capacities up to eight megabytes. While eight megabytes is much larger than floppies available at that time, it was not adequate to handle hard disks of several dozen megabytes.

One way to work with this limitation was to declare each physical hard disk to be several logical disks. The CP/M disk parameters included a field originally intended to be used to say how many tracks were reserved to keep copies of the CP/M itself. These reserved tracks started at the outer track, and continued up to the last track to where the directory is started. By using the correct values for the offset parameter, each hard disk can be divided into bands, with each band starting inside the area used by other logical drives.

Thus, to make a ten megabyte drive with 900 tracks able to emulate two logical floppy disk drives, the first drive would be given an offset of 0 (unless the system was to be stored on the hard disk), while the second logical drive would start at track number 450.

CP/M Plus (CP/M 3.0) further expands the maximum amount allowed per disk to 512 megabytes, quite adequate for the forseeable future in small hard disks. Each file can be a maximum of 32 megabytes.

A hard disk can turn a small system into a large one capable of storing and retrieving larger amounts of data at faster speeds than with floppy disks. The price is falling, the capacity is increasing, and the technology is ripening, but now may be the right time to upgrade your personal computer with a hard disk.

Software For Hard Disks

With so much room on a hard disk, good organizational techniques are essential for keeping track of your programs and projects.

User areas, as described in a CP/M Plus tutorial in this issue, let you separate files into logical areas on disk within each directory. While logged into one user area, you don't see the files in the other user areas, although with CP/M Plus you do have access to files in user area 0 marked with the system attribute.

User areas and other features of CP/M Plus, such as the time and date of each file's last update, also come in handy most when you're using a hard disk.

Letters...

Continued from page 7

at. It always adds two or three lines and then you can never get the page lengths to come out right or you have to super guess where it will begin filling in a form. Is there a way of printing on the line up line and still get page numbers?

Also, I have a Diablo 1610 that just thrives on ETX/ACK protocol. It also has a 1200 baud option that works about two times as fast as the 300 baud option if the ETX/ACK protocol works. WordStar has a "Catch 22" problem with CP/M. If your BIOS has the ETX/ACK protocol so that you can print properly, then you can't use WordStar's ETX/ACK protocol.

That might seem great except WordStar sends ESC sequences to do fancy printing that are very difficult to program for in the BIOS. If you use WordStar ETX/ACK, then you can't use the BIOS ETX/ACK because they are mutually exclusive. If you don't mutually exclude them, one always swipes the other's ACK. If the BIOS does not have the ETX/ACK protocol, then you cannot *TYPE* a file out using CP/M's Control-*P*. That is very painful.

Obvious solutions to the problem involve a separate WordStar disk, but this is such an in-elegant solution that I don't even want you to think I said that. What about the possibility of a second driver using the extra port thing in WordStar? I bought WordStar's Customization Notes but they were no help.

Sure am looking forward to my next *User's Guide*.

Sam Lay
S C Management Consultants
Costa Mesa, CA

Ouch! ("Which catch is that?" Catch-ETX/ACK). There is (almost) no hope unless you are willing to drive the printer directly from WordStar using those port locations. This involves finding out which port services the printer... perhaps a local technician can help. We intend to provide a WordStar customization note ourselves... — Editors.

With reference to Page 23 of your issue no. 1, "Use the Stat Command to give the Sys attribute to programs that are needed by other users...and store...in user area 0..."

We, and others find we are not able to run SYS programs in user 0 from other user areas. What's wrong? Our machines are Osbornes and Northstar.

Many Thanks.

Bob Battes
Lockheed MSD
Sunnyvale, CA

Sorry to say this, but the feature described works only in CP/M-86 and CP/M Plus systems, not in CP/M version 2.2 or earlier. — Editors.

Issue #4 was most interesting. At some point though, I hope you will cover CP/M-86/80 designed for DEC Rainbow 100. It seems to have features like MAINT and editor RED not found on other CP/M systems.

Also, I join those who await your review of income tax for SuperCalc.

Finally, you should add the DEC Rainbow to your list of hardware for future reviews. DEC's marketing ploys have not helped, but after three months of use, I find it an excellent machine.

Derrick Bell Eugene, Oregon

An article on the DEC Rainbow will appear in the next issue. Unfortunately we have not received appropriate tax templates for SuperCalc, and we are also still waiting! — Editors.

Can you provide me with the addresses of CPMUG and SIG/M for the distribution of 8-inch disks of the public domain software?

SIG/M charges $6 for the first disk and $5 for additional disks, and the address is SIG/M, Box 97, Iselin NJ 08830 (eight-inch single-density single-sided disk only). CPMUG charges $13 per eight-inch disk, and the address is 1651 Third Ave., NY NY 10028. — Editors.

TeleVideo 950 Terminal & WordStar EPROM Kit

Performing WordStar® Commands Using the Function Keys of a TeleVideo 950

by Tony Bove & Cheryl Rhodes

We had success with the TeleVideo 950 WordStar function key modification. We're writing and editing this piece with the TeleVideo 950, using the function keys as intended; however, as veteran WordStar users we find it hard to put aside our old habits, especially when typing fast. A WordStar beginner we trained on the TeleVideo 950 with the WordStar EPROM modification found WordStar easy to use.

TeleVideo Systems, Inc. (Sunnyvale, CA) manufactures whole computer systems as well as terminals. The TVI Model 950 is a high-performance ASCII video display terminal (VDT) with "smart" features: 11 function keys programmable for 22 functions, a variety of screen display and editing features (including line-erase, page-erase, smooth scroll) and send/receive and printing capabilities in local mode. A typical "smart" terminal with detached keyboard for a reasonable price — $1195.

The 950 WordStar/CPM EPROM kit is optional and consists of three tiny EPROM (erasable programmable read-only memory) devices and installation instructions. Installed, the EPROMs let you switch the function keys to represent WordStar commands or CP/M version 2.2 commands.

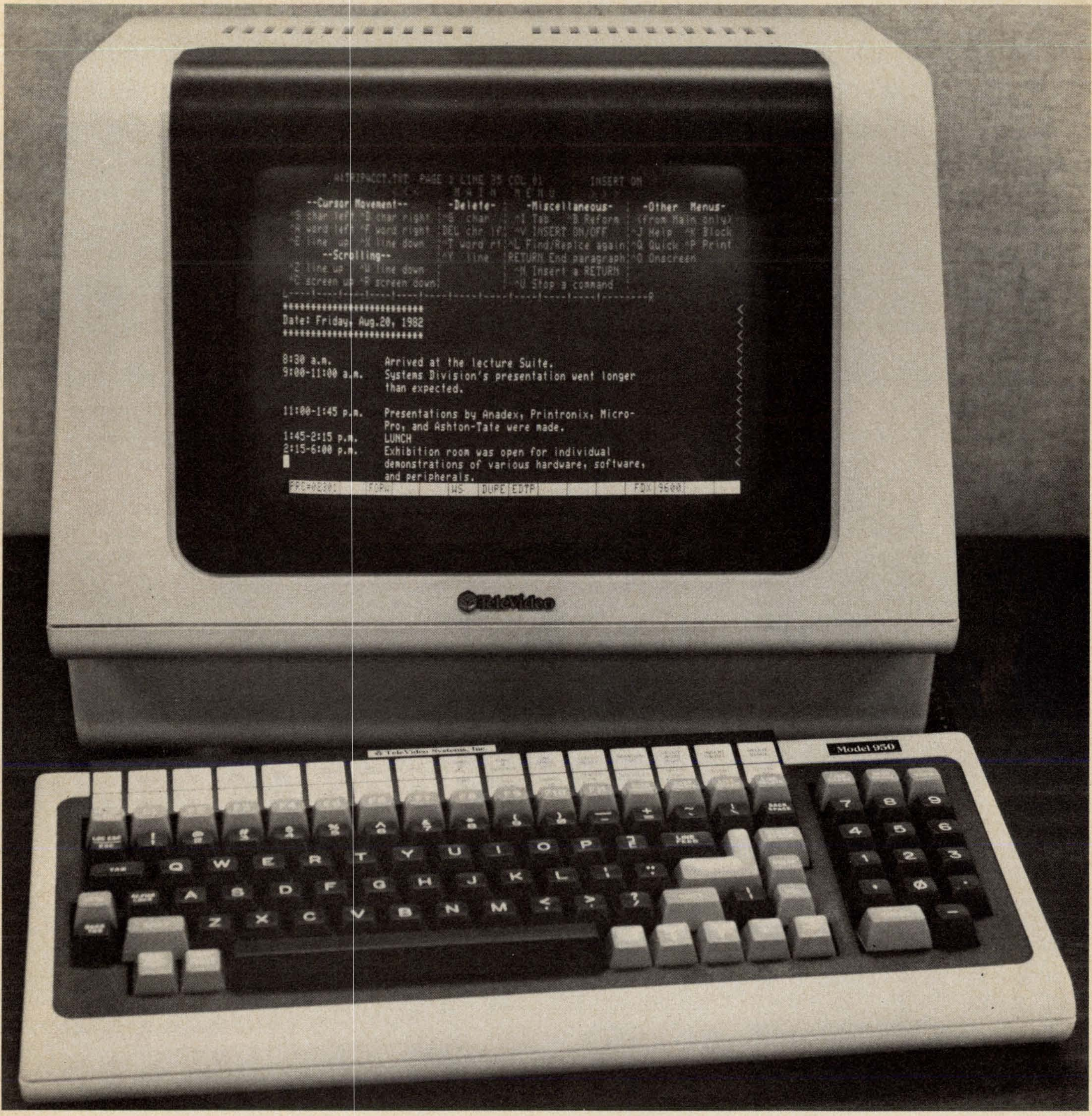
WordStar Habits Hard To Break

We've used WordStar for over four years, and our nit-picking may be too rough on this product, so we trained a WordStar beginner using this terminal with the EPROM modification. The result was that the beginner found WordStar very easy to learn, contrary to *most other* beginners' experiences with vanilla WordStar. The function keys were obviously WordStar commands, whereas the Control-key combinations (what we're used to) were not so obvious.

WordStar habits are hard to break. Our initial nit-pick was the lack of a function key for moving the cursor quickly to the end of a line. The modified 950 offers functions like "Jump to End" and "Jump to Beginning" (two things we do often), but no "Jump to end of line" or "Jump to beginning of line." WordStar's ↑*QD* jumps to the end of a line, and ↑*QS* jumps to the beginning; and with the "Q", "S" and "D" keys right near each other and also close to the Control key (displayed as ↑), the WordStar commands are easy to type. Once we started again using the old WordStar commands to move the cursor, it was hard to remember to use the new function keys, and we found ourselves reverting back to our old habits.

We must admit, the CHAR DELETE (character delete) and REFORM PARAG. (reform paragraph) keys were very useful. However, the LINE INSERT and LINE DELETE are too close together. It should be harder to delete a line — people do it by mistake very often in WordStar. A friend who owns a Televideo 950 removed the key cap from the LINE DELETE key. Now he only deletes lines when he wants to, because he can feel the difference between LINE INSERT and LINE DELETE.

We have not yet adjusted to the rapidly-repeating feature of the keyboard, and when we repeat an arrow key to move the cursor, we nearly always overshoot the target. We have not figured out a way to adjust this "feature" of the 950 terminal. Perhaps others who like the rapidly-moving arrow have better hand-eye coordination.



```
     A:TRIPACCT.TXT  PAGE 1 LINE 35 COL 01        INSERT ON
                          M A I N   M E N U
  --Cursor Movement--     | -Delete-  |   -Miscellaneous-   |  -Other  Menus-
^S char left ^D char right |^G  char  |^I Tab   ^B Reform   | (from Main only)
^A word left ^F word right |DEL chr lf|^V INSERT ON/OFF     |^J Help  ^K Block
^E line  up  ^X line down  |^T word rt|^L Find/Replce again |^Q Quick ^P Print
     --Scrolling--         |^Y  line  |RETURN End paragraph |^O Onscreen
^Z line up   ^W line down  |          | ^N Insert a RETURN  |
^C screen up ^R screen down|          | ^U Stop a command   |
L----!----!----!----!----!----!----!----!----!----!----!---------R
**************************                                          <
Date: Friday, Aug.20, 1982                                          <
**************************                                          <
                                                                    <
8:30 a.m.          Arrived at the lecture Suite.                    <
9:00-11:00 a.m.    Systems Division's presentation went longer      <
                   than expected.                                   <
                                                                    <
11:00-1:45 p.m.    Presentations by Anadex, Printronix, Micro-      <
                   Pro, and Ashton-Tate were made.                  <
1:45-2:15 p.m.     LUNCH                                            <
2:15-6:00 p.m.     Exhibition room was open for individual          <
█                  demonstrations of various hardware, software,    <
                   and peripherals.
PRC=02301   FOPW        WS  DUPE EDTP              FDX 9600
```

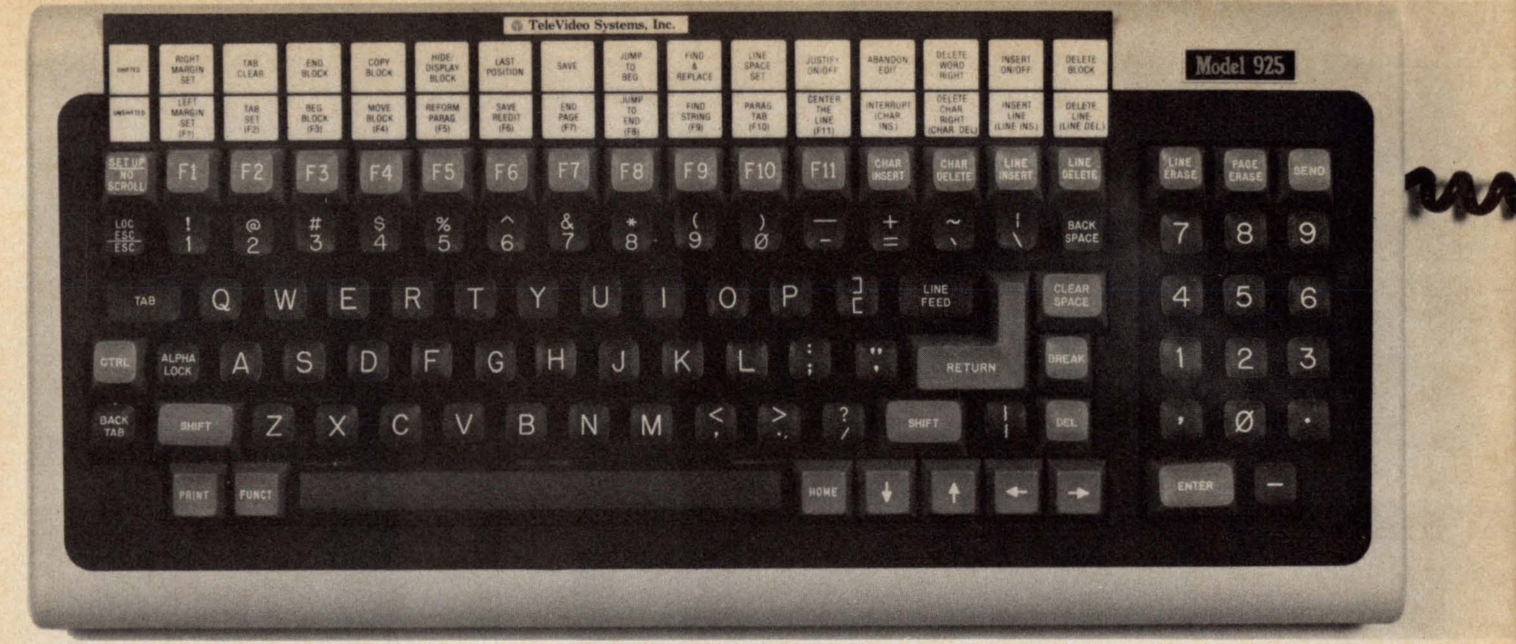
Favorite Commands and Suggestions

Other favorite commands we wish to see included as function keys are the "Move Forward/Backward By Word" functions. They are easily accomplished with ↑*A* (backward) and ↑*F* (forward), but once again, we are back to using normal WordStar commands and not function keys, and once we have our finger on Control...

We use these commands often while writing (hmmm, don't like the sound of that, I'll move back and insert an adjective — ↑*A* back a few words, type a few words, then zoom to the end of line with a ↑*QD* or zoom a few words ahead with ↑*F* ... Haven't used any function keys yet...).

The rapidly-repeating feature of the keyboard again revealed itself when we accidentally typed the backtab key while trying to find the shift key and found ourselves scrolling rapidly line-by-line forward through the file. This can only happen when you are in WordStar mode. You toggle between CP/M and WordStar modes with an ESC and SHIFT and ⮐

In addition to the backtab key placement, other extraneous, unusual



"action keys" include PRINT and FUNCT (left of the space bar), and HOME and the four directional arrow keys (right of the space bar).

We looked over the keyboard layout and decided to suggest one change: Shift with a left or right arrow key could move the cursor to the beginning or end of a line, like Scripsit on the TRS-80 Model III (funny... the keyboard finish also reminds us of the Model III keyboard).

We had a few problems. The F7 function key is supposed to insert a .PA in the file to end a WordStar printed page. We tried unsuccessfully to make this key work. It would insert only the period (.) of the command, not the whole command.

To be fair, this function key setup was designed for beginners. Everyday WordStar commands are presented in a way that is easily understood. Since learning WordStar tends to get in the way of writing, the TeleVideo 950 is a terminal for WordStar beginners to use to learn new habits.

The best feature is the "Save and Re-edit" function key, which makes sense — push a button to save the file. You use one key to save, and you SHIFT the same key to return to where you left off.

Installation

Installation was not as easy as described in the skimpy installation instructions. It was easy to take off the cover, but the logic board did not look like the one pictured in the manual — there were major differences.

The instructions called for sliding the entire logic board "out of the terminal." However, the logic board would not budge without considerable tugging. The instructions advise that you pull up on the corners to avoid scraping the underside of the logic board on the chassis of the terminal when sliding it out. We found it very hard to avoid scraping the underside not just at the corners, but across the entire board.

Beginners should not install this modification unless they know how to handle EPROM components.

The Keyboard Is Everything

The keyboard has an excellent feel and response. This is extremely important to touch-typists like ourselves. Fast typists require this kind of response, and many other terminals do not deliver it.

Although the function keys on the 950 help teach WordStar functions, the keyboard can also frighten beginners. Add the function key description label to the keyboard, and it becomes more complex. At first glance there are too many keys to recognize and too many functions to understand.

On the other hand, typists who are used to dedicated word processing machines will not be intimidated by the 950 keyboard, and they will learn WordStar quickly, because they are used to using function keys for commands. The templates that explain the keys are similar to templates on word processing keyboards, and they are useful as memory joggers and learning aids.

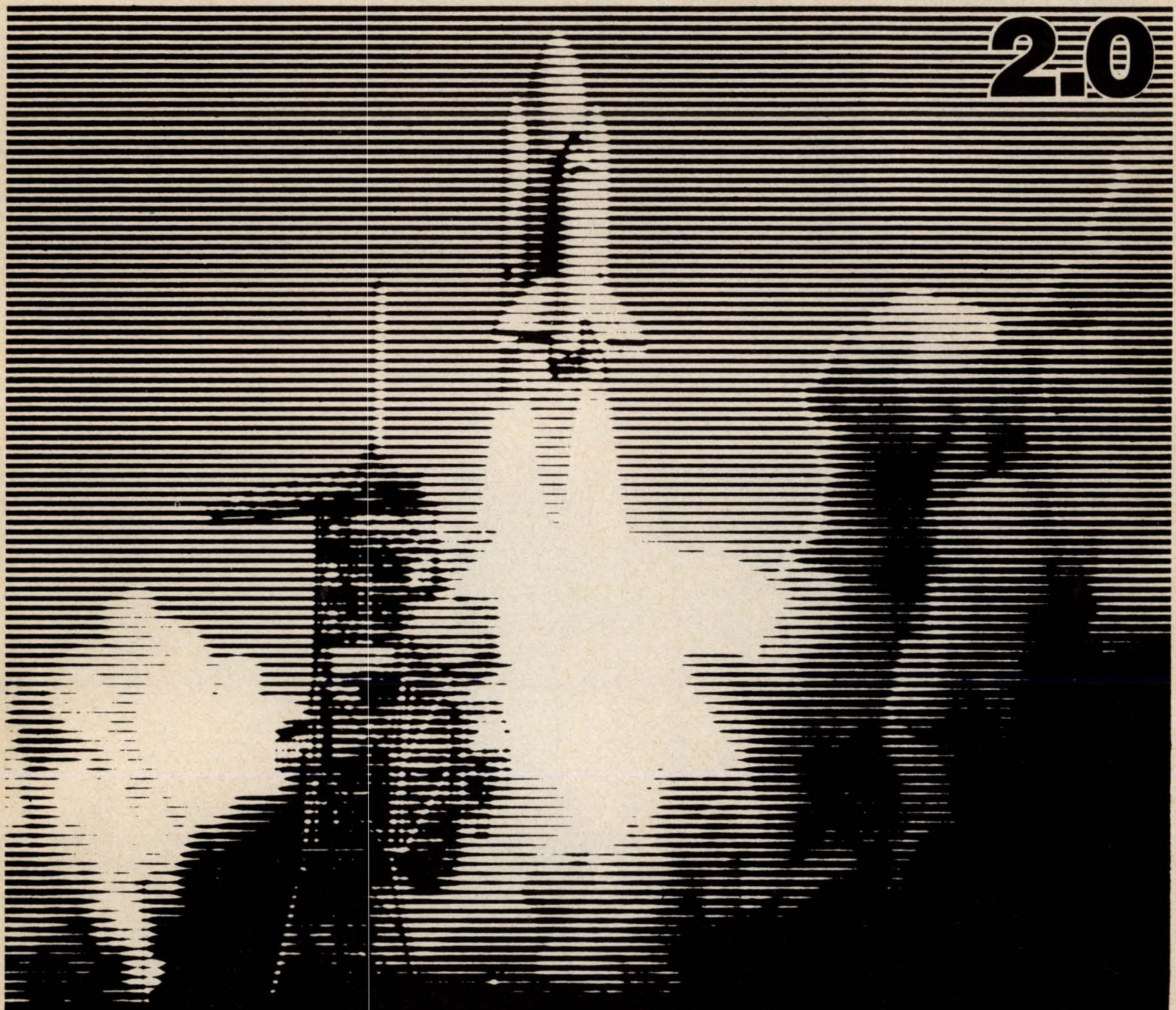
WordStar is an excellent tool for fast typists. Touch typists hover their fingers over the "ASDF" and "JKL;" keys, and WordStar provides functionality in those keys, with the help of a "Shift-like" key — the Control key. The EPROM-modified TVI 950 works well in training mode, but many fast typists might switch to using the Control-key commands for positioning, and use function keys for some of the commands.

Many touch typists we know, including ourselves, dislike having to leave the typing area of the keyboard to press a function key. In long hours of typing, use of the Control key can get painful. In short "blasts" of writing, a momentary digression to another part of the keyboard can cause a panic in the creative thought flow.

It takes time and practice to get used to using any new terminal. The best judge of a terminal may be someone who likes to use it despite its difficulties. Terminals and keyboards come in different styles and shapes, but once you decide how much you want to spend the choices are usually narrowed down by these qualities: good display, good keyboard layout, and good keyboard feel. The TeleVideo 950 does win on all three, but the WordStar EPROM modification could use some flexibility for user changes, and better documentation.

Nothing On Earth Compares With

DATAFLEX™



Incredible power, un-precedented speed, and application flexibility for virtually any mission. That's DataFlex!

And now, the new DataFlex command language is the friendliest way you can deliver your software payload. All of the extraordinary state-of-the-art features of DataFlex 2.0 make it the clear choice for an applications data base.

But there's an even better reason to buy it... it grows with you. Needs change, your single user system may expand to multi-user, new requirements need to use valuable information developed in present applications. DataFlex will never keep you down!

DataFlex...available in single user and multi-user versions for 8 and 16 bit systems with extensive support for local area networks.

Get into orbit now with DataFlex. The applications environment of the '80's.

DATAACCESS
CORPORATION

8525 SW 129 Terrace
Miami, Florida 33516
(305) 238-0012 TLX 469021
Data Access CI

Circle number 5 on the reader service card.

System Overview...

The Radio Shack® Model 100 Portable "Lap" Computer

Suitable for Writing, Note-taking and Daily Messages, The Model 100 makes a Useful "Satellite System"

by Dan Dugan

© 1983 Dan Dugan

TRS-80 is a registered trademark of Tandy Corp. (Ft. Worth, TX)
WordStar is a trademark of MicroPro, Int. (San Rafael, CA).

(c) 1983 Rose Skytta

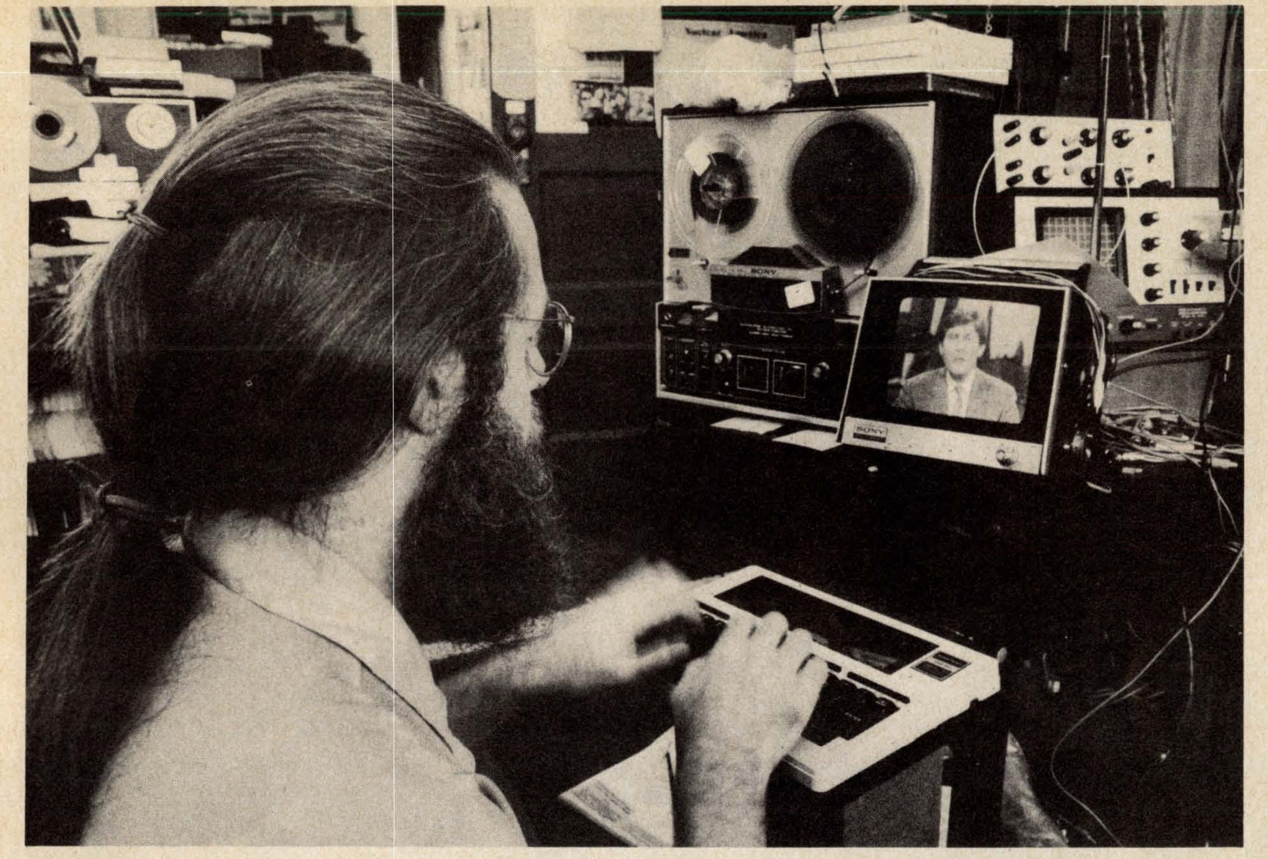
I couldn't resist it, I've been waiting for it for years. A word processor with a real keyboard that sits in your lap. I saw it first on the cover of *Computers and Electronics* (May 1983)... hands trembling, I turned to the article, read it in one gulp, took the Metro down to Radio Shack and bought a TRS-80 Model 100.

In just a few weeks, several helpful applications emerged. One of the best is using it for keeping my priority list, a list of phone calls and things to do. To store the latest messages I open my file P.DO and type in phone numbers at the top of the list. A couple of times a day I use the cursor keys with the cut and paste functions to move tasks up and down the list as priorities shift.

I type my journal every day using the Model 100, and I can keep it up to date more easily. For several years I typed the journal at a terminal, but who wants to sit up straight at the end of a day? Now I type my daily notes in bed, a real improvement.

An extremely useful application well-suited for the Model 100 is logging recording notes. I tape-record segments from broadcast news, and for years I've kept notes of the contents of the tapes on binder paper. Now I keep them in a text file, uploaded periodically to a hard disk. In machine-readable

(c) 1983 Rose Skytta



form these files can be searched more rapidly for names or topics.

The Model 100 is a godsend for inspired writing. Unfortunately the limited storage gets filled up pretty quickly. I use it as a satellite to a disk-based system, transferring files back and forth.

The Model 100 can be bought stripped-down for around $800, but for a real system (with the full 32K of user memory and the required phone and cassette cables), the price is closer to $1300.

I strongly recommend buying the full 32K memory. Consider that used Osbornes are selling for that price; the machines are complementary rather than competitive.

Think about this: right now, for around $2600, you can buy a CP/M computer system with at least 320K of floppy disk storage (the PMC Micro-Mate, used Osborne or used Kaypro), *and* a satellite 32K memory lap computer/terminal (Model 100). I expect to see a lot of wonderful little lap machines like this coming out, and I expect prices to plummet, if you can wait.

The Screen and Keyboard

The screen is liquid-crystal, with eight lines of forty characters. The inability to scan the whole text rapidly, already a disadvantage on a regular 24-by-80 screen, is much worse on the 8-line screen. Given this limitation, there is enough context visible to make writing comfortable.

The screen characters are large and easy to read, but the display is very critical of lighting and reflections. It is necessary to adjust a unique tuning knob to make the display visible from the viewing angle — the image disappears when not tuned. The angle of the light isn't critical but is important for comfortable viewing. Lights or bright images above and in front of the user will reflect off the screen and reduce contrast. An ill-designed black mask around the screen window often casts a shadow obscuring half the top line or the first column on the left.

The keyboard is extremely comfortable and fast. It's near-perfect, lacking only a palm-rest and tactile home-row key identification. The keyboard must be good for this type of machine, because one tends to use it at odd angles.

There are 16 little function keys across the top of the keyboard, eight of which are programmable and take on different functions in different programs. The functions of the keys are identified by labels in the bottom (8th) line of the screen. The screen scrolls in the remaining 7 line window when the labels are on. Poor design positions these keys so that they don't line up with the on-screen labels.

In addition to the control shift, there are also GRPH and CODE shift keys, which allow the keyboard to generate every one of the 256 possible characters (eight-bit bytes). These are used for a large library of graphic elements and symbols, many useful, some mystifying.

Designers of lap computers could

Software

When the computer is turned on it displays a menu of programs and files, along with the time and date from the internal clock-calendar. Moving the cursor to the name of a text file and pressing ENTER will open that file for text editing. No program loading is required. Moving the cursor over any built-in or BASIC program's name and pressing ENTER will start that program.

A feature that puts this machine into a new generation is the automatic saving of anything typed in. You never have to give a "save" command! Well, almost never. It's necessary to save a new BASIC program once to give it a name. Subsequently, anything added to the program is automatically saved. Anything written in a text file will be there when you come back. It is a great relief to be able to concentrate on the task at hand and be assured that work will not disappear.

Writing

The word-processing program is simple but consistent in its logic, and should be easy to learn by anyone. As an experienced word-processing person who uses WordStar®, I was pleased to see the familiar cursor-control keys. There are minor variations, and I miss the fancier things that WordStar can do, but the Model 100 does everything that is necessary, including searching for a word or phrase. There is a very easy cut-and-paste operation for moving any-sized blocks of text, even between different files.

The display is fast when typing at the end of a file. When inserting text within a file there is a delay while the program moves the text down. The computer buffers up to 40 keystrokes and displays them as fast as it can, which may be a few seconds if a long burst is typed in the middle of a file.

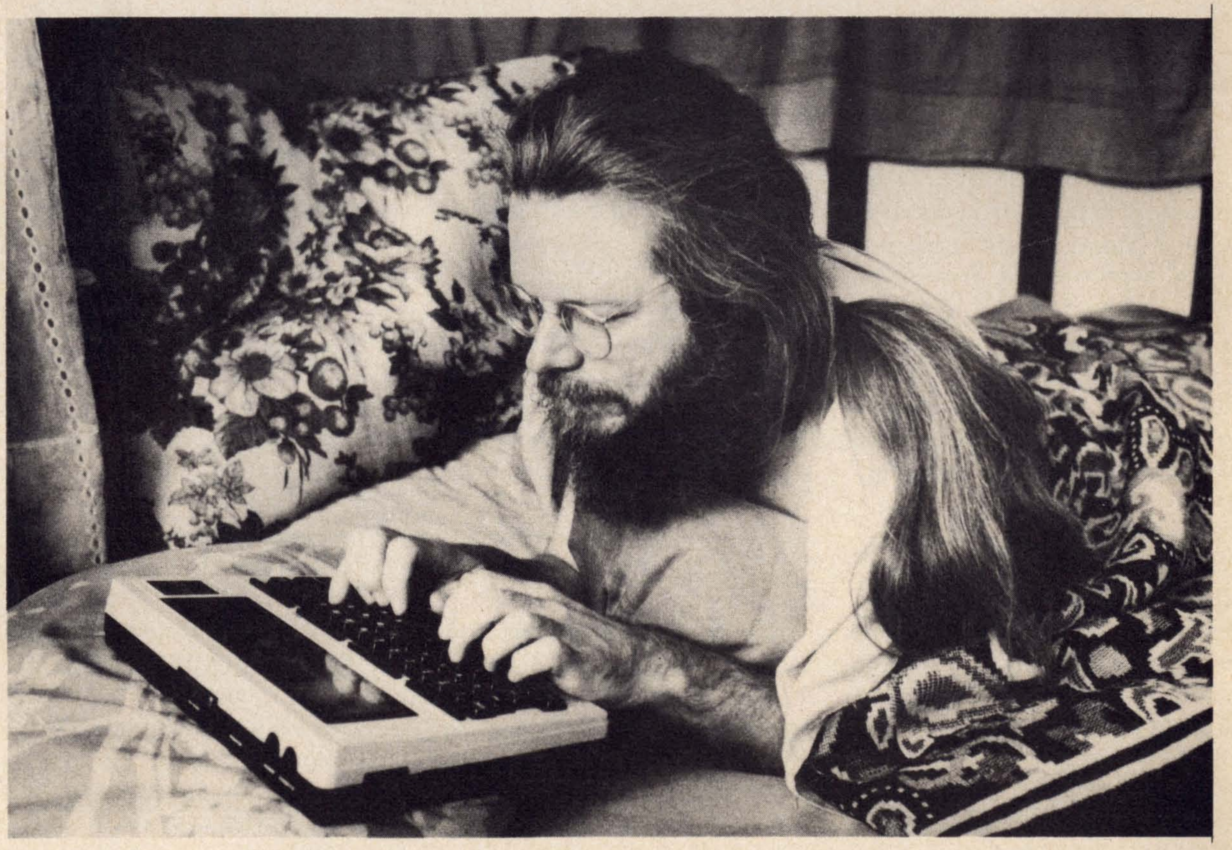
Telecommunicating

After writing and editing, the next most valuable software feature of the Model 100 is the TELCOM program, which operates either the built-in modem or the RS-232 serial port.

The computer plugs into the world in two ways: through the built-in modem to the telephone network, or through the serial port to another computer, terminal, modem device, printer, or other serial device.

The built-in modem, with the modular phone cable (sold separately), can turn the Model 100 into a terminal on any remotely-operable computer system. Bulletin boards, remote CP/M systems (RCPMs), and commercial information services (such as CompuServe or The Source) can be accessed with ease.

The program includes automatic pulse-dialing, and can be programmed



(c) 1983 Rose Skytta

learn a lot from using the Radio Shack Model 100.

to dial numbers and send sign-on replies and passwords automatically.

For example, to access CompuServe I turn the power on, press the space bar twice to position the cursor over TELCOM on the menu, and press ENTER; then I press F1 (for "find"), type *compu*, and check to make sure it found the right number; finally I press F2 (for "call"). I watch while the program dials the phone, signs me on and delivers me to the CompuServe menu.

Uploading and Downloading

TELCOM can also communicate directly by serial cable with another computer and exchange files. Getting this procedure set up the first time between the 100 and any computer probably requires the assistance of a person experienced in the ins and outs of hardware and software communicating, but once the routine is figured out shuttling files is a just a chore.

(We cabled the Model 100 to the PMC MicroMate, and used the Model 100 as a terminal to control the MicroMate computer. This gave us access to at least one 320K floppy disk drive. See the article about the PMC MicroMate in this issue.)

I've just been "on location" for three weeks working on a Shakespeare festival, and I took the Model 100 along for taking notes. I also brought my Osborne 01 portable computer to serve as a "mother ship" — to receive and store files on floppy disk as the Model 100 ran out of storage. The Model 100 needs a program to make uploading and downloading batches of files easier.

You can switch the input and output of TELCOM to either the built-in modem or the RS-232 serial port, which has a standard DB-25 plug. The modem is fixed at 300 baud, but the RS-232 port can be set for a wide range of speeds (75 to 19,200 baud).

The built-in "download" command is only usable at speeds over 300 baud if the host computer responds to X-ON, X-OFF protocol. This is because it displays everything received on the screen, and otherwise the host doesn't know when to wait for scrolling. The "upload" command asks what line length to send and then formats text into lines with carriage-return characters. This causes problems with text editing programs on CP/M systems, which expect lines to end with both the carriage-return and line-feed characters.

To solve this problem, I wrote a simple BASIC program (*see insert*) to send files out through the serial port as paragraph-long lines, so that I can format the file to any width with WordStar on my Osborne or office system.

While mentioning ins and outs, there is also a parallel printer connector, for which a cable is sold separately, purported to be Centronics compatible (meaning that Epson printers, etc. should work). I haven't tried it.

There is also a connector for a cassette recorder cable (sold separately) that works with any standard cassette. I tested loading and saving files from cassette and it worked well after very little fiddling. The output to the tape is line level (-10 dBm) and probably too hot for recorders with only a mike input jack.

Programming

As a programmer, I am delighted with the built-in BASIC language, designed by Microsoft, which is nearly a full implementation of BASIC-80, plus extras for the Model 100. It has no random files and no function definitions, but it has multi-dimensional arrays. A very flexible file system gives full control of RAM files including APPEND.

Another programming advantage not available in the CP/M version of BASIC-80 is the ability to address the screen display, cassette interface, modem or RS-232 ports as files from within programs. Although variable names are restricted to two characters, remember that this machine is designed to work with limited storage space.

A really spectacular feature of the Model 100 BASIC is the interrupt capability. By using the program statement *ON KEY GOTO ...*, the function keys can be programmed to execute BASIC subroutines from any point in the program, returning to wherever the program was when the key was hit.

Similar statements can interrupt a program on error, at internal clock times, or when a character arrives at the modem or RS-232 port. This offers rich creative possibilities. The computer can turn itself off automatically (but the ability to "wake up" is unfortunately missing).

The BASIC includes graphic commands. The cursor can be positioned to any screen location, for example *PRINT* 140. Individual pixels can be turned on and off — e.g., *PSET(239,63)*. Lines and boxes may be drawn with built-in commands, including box filling. The fixed 256-character set includes many useful and many strange European-accented and graphic characters. The special characters are not ASCII, since they use the 8th bit, and may be lost if uploaded to a CP/M machine.

Essentials

The Model 100 comes with TEXT (a text editing program), BASIC, TELCOM, a built-in modem, up to 32K internal memory, one RS-232 port (cable extra), one DIN telephone jack (modular adapter cable extra), and one DIN cassette connector (adaptor cable extra). Included are connections for future expansion: a nine-pin connector for a bar code reader, and inside a hard-to-remove panel on the bottom, an empty ROM socket and a bus expansion socket. Pin-outs for all connectors are in the manual.

A soft carrying case is included. The price (8/83) is $799 with 8K memory, $999 with 24K. The 8K internal memory expansion is offered for $119.95 plus labor.

Information

Two good manuals are provided: a spiral-bound 224-page users manual having both step-by-step tutorial and reference information, and a comprehensive pocket-sized 50-page "Quick Reference Guide." The only major mistake I've found in the manuals is the definition of the BASIC TAB function, which tabs to a specified column in the standard way, fortunately not the odd way defined in the manuals.

There is an accumulation of technical information of interest to assembly-language programmers available in the programmers' SIG on CompuServe.

Power Requirements

The Model 100 runs on 4 AA cells, just like a Sony Walkman®. Alkaline batteries lasted about three days, with plenty of reserve after the "low battery" light comes on. The unit turns itself off after a programmable period of inactivity; I prefer it at 3 minutes (set by the BASIC command *POWER 30*).

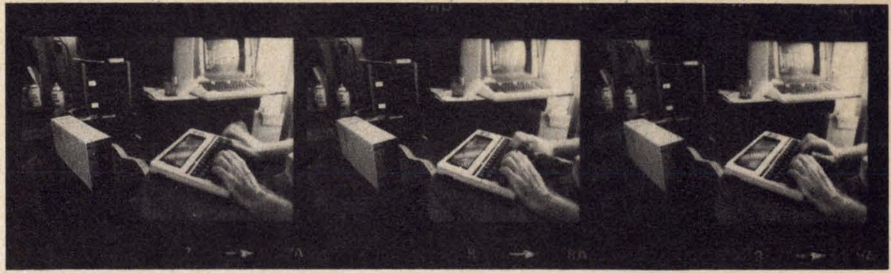
I purchased a "Dynacharger" and several sets of nicad cells for it, and these last more than two days of heavy use. Unfortunately, the low battery light comes on when the nicads have no reserve at all — meaning, change batteries right now. Nothing is ever lost by running out of power, since there is a hidden internal nicad battery that keeps the memory alive when the power is off, for days. I use three marked sets of nicad cells, one in the machine, one travelling with me ready to switch and one set in the charger.

Overall Feeling

Designers of "notebook" or "lap" computers could learn a lot from using the Model 100. It is harder than I would like to carry it around — it's heavy, the edges are sharp, and there is no handle. I'd like to have a built-in handle like the one with the TI Speak-'N-Spell®, but below the keyboard and bent upward so that it would also provide a palm-rest and protect the keyboard when the unit was face-down.

A "double density" screen (16 x 80) is inevitable in the future; however, the 8 x 40 character screen is the best that's available to manufacturers currently. As you can see in these photos, I have found many uses for it, and I feel comfortable using it.

(c) 1983 Rose Skytta



Uploading To CP/M and WordStar

A Quick and Useful Transfer Program For the Radio Shack Model 100

by Dan Dugan

The program's name is PUTFIL.BA (BASIC programs have the ".BA" extension). It is a utility program for sending files to PIP or other CP/M programs. It sends "paragraphs" that can be re-formatted into WordStar paragraphs.

```
30 CLEAR 2000                 'for paragraph
40 MAXFILES=2
```

(change the values of the following three lines to meet your needs).

```
42 NU=0                       — set nulls 1=on (temporary fix for problem interfacing
                              with some other computers' RS-232 ports).
44 Z=1                        — set ↑Z's at end 1=on (needed to tell PIP you're done).
50 CO$=''88n1e''              — set default baud rate to be used when you enter just
                              "com:"in response to the destination question.
52 CLS
60 PRINT''PUTFIL
70 FILES
80 INPUT''Filename to read (or Menu)'';F1$
90 IF F1$=''Menu'' THEN MENU
100 OPEN F1$ FOR INPUT AS 1
110 INPUT''Name of file to write (or com:[rwpbs?])'';F2$
120 IF F2$=''com:'' THEN F2$=F2$+CO$
130 OPEN F2$ FOR OUTPUT AS 2
140  IF  LEFT$(F2$,4)=''com:''AND  NU  THEN  FOR  I=1  TO
50:PRINT#2,CHR$(0);:NEXT
150 IF EOF(1) THEN190
160 LINE INPUT #1,L$
170 PRINT #2,L$
180 GOTO150
190  IF  LEFT$(F2$,4)=''com:''AND  Z  THEN  FOR  I=1  TO
128:PRINT #2,CHR$(26);:NEXT I
200 CLOSE
210 GOTO60
220 END
```

I found that my computer's RS-232 port worked fine with one computer, but garbled characters with another. This was due to a design error in the Model 100. Radio Shack has corrected the problem and older computers having this problem will be fixed free - ask the service center to check Technical Bulletin 100:01.

For the next 30 days only we will match
any price in this issue of User's Guide
on any hardware or software products
we now have in stock

There is only one number to call:

1-800-237-8977

In Florida call 1-813-786-1259

We will also give you an additional 10% off
that advertised price just to introduce
you to our toll-free total support.

The Systems Group

2340 State Route 580/Suite 234
Clearwater, Florida 33575
Telex 701484

There is only one catalog to have:
our new, **free** 100 pages of the best
in hardware and software. Send for
your copy today!

IBM is a trademark of IBM Corporation
APPLE is a trademark of Apple Computer Inc.
TRS-80 is a trademark of The Radio Shack Division of Tandy Corporation
XEROX is a trademark of Xerox Corporation

Offer expires Oct 15, 1983.

Circle number 15 on the reader service card.

The PMC MicroMate With CP/M Plus: A Transportable and Affordable System

A Computer For All Terminals, From Workstations To Portables

Reviewed by Tony Bove & Cheryl Rhodes

In the wake of an armada of product announcements for new sixteen-bit computers, a flotilla of the latest eight-bit computers entered the market, hoisting the CP/M Plus flag. Eight-bit technology has matured and become even more useful for businesses and professionals.

Eight-bit computers can now grow to 128K or more in internal memory, and manage that memory with the new "banked" version of CP/M. The PMC MicroMate is an inexpensive entry into this market, running 128K CP/M Plus and industry standard software.

The PMC MicroMate is *transportable*, weighing in at only eight lbs., and is enclosed in a box that also houses a slim 5.25-inch floppy disk drive. The unit is 3.5" wide, 6" high and 15" deep, and fits in most briefcases. More disk drives can be connected to it. The MicroMate does not provide a built-in screen or keyboard; however, any ASCII-type terminal can be used with it.

It is light and can be used on any horizontal surface. You can hide it behind a regular-sized CRT terminal and pretend to be connected to a large computer system. We expect that many business people who use terminals in their offices will find the PMC MicroMate to be the perfect add-on for distributed processing.

Use your favorite keyboard and screen. You can change terminals quickly, since the MicroMate gives you software control over the serial ports from CP/M Plus. You can use any ASCII terminal or personal computer operating as a terminal. That can include anything from the finest workstation on a large computer system to the smallest pocket terminal that speaks in ASCII. We used the PMC MicroMate with an ADDS Viewpoint, a Televideo 950, a Qume QVT-102, and a lap computer (the Radio Shack Model 100).

Essentials

The unit retails for $1195, which includes CP/M Plus, CBASIC, Electronic Webster spelling checker, and T/Maker III, one of the finest integrated packages on the market (regular value of T/Maker III itself is $275).

The basic unit (model PMC-101) does not come with a display screen or keyboard (for another $740 you can get the Qume QVT-102 with cable included). It has two serial ports (50 to 19,200 selectable baud rate): one for a terminal and one for a modem. It has a parallel expansion port for disk drives and a parallel Centronics-type printer port (for Epson and other printers). There is also an internal bus connector providing high-speed parallel I/O channels for up to 128 devices.

One MicroMate-format floppy disk holds 400K, and you can connect directly to the computer more PMC boxes that have one or two external floppy disk drives, for a total of 1.6 megabytes in four drives.

The floppy disk drives have another essential ingredient missing from most computer systems: the ability to read and write to other types of disks. The MicroMate can use disks from IBM PC, Obsorne, Xerox, Kaypro, Omikron, Morrow, H-P, DEC, Cromemco, InterSystems, Televideo and Zenith personal computers. As far as we know, this list of compatible disk formats is

MicroMate is a trademark of Personal Micro Computers, Inc.(Mt. View, CA).
Radio Shack is a registered trademark of Tandy Corp. (Fort Worth, TX).

(c) 1983 Rose Skytta

the largest for any single computer using 5.25-inch drives.

Overall Feeling

A reviewer of the Radio Shack Model 100 (this issue) uses the lap computer when traveling, and brings along an Osborne to act as a "mother ship" with disk drives. He uploads text to the Osborne to store on disk.

We tried his idea substituting the MicroMate for the Osborne. We brought the Radio Shack Model 100 to double as a lap computer *and* a terminal for the MicroMate. The combination fit neatly into a large briefcase. The MicroMate's double-density double-sided disk format holds up to 400K, which is more space than two 192K Osborne disks. Also, the MicroMate's drive can read many other disk formats, including Osborne and Kaypro.

Since many inexpensive computers have flaws in their screens or keyboards, we welcomed the flexibility of using the MicroMate with any terminal. Anyone who has tried to write on a computer knows about the advantages of using a good keyboard rather than a mediocre one. To some of us, the keyboard is the most important consideration when buying a computer.

With unlimited choice in terminals, the MicroMate is cost-effective for anyone who uses a terminal or personal computer in the office and needs a compatible computer at home to use with an inexpensive terminal.

The MicroMate is one of the least expensive CP/M Plus computers. It offers the banked version of CP/M Plus (3.0) with 128K of internal memory. All-around performance is better than ordinary CP/M 2.2 systems because the memory management of CP/M Plus with bank switching greatly reduces disk access time.

CP/M Plus also provides new features not available in ordinary CP/M 2.2, such as password protection for files, command line editing and commands to direct printer output to a file or retrieve keyboard input from a file. The password protection and time/date file recording features reduce the complexity of managing a hard disk, which is the perfect companion for the MicroMate. PMC offers a 10 megabyte hard disk for the MicroMate.

For most business and professional applications, we recommend plenty of disk storage. You can connect another one-drive unit, or connect a two-drive unit for a total of three (1.2 megabytes), or connect both extra units (for a total of four drives and 1.6 megabytes).

A 10 megabyte hard disk with the single drive could provide all the storage you need, if you don't mind performing file backup to the single 400K floppy drive. For word processing (involving lots of small text files), this could be the ideal and least expensive CP/M Plus system.

Information

One manual specific to the PMC MicroMate is supplied along with the entire boxed set of CP/M Plus manuals from Digital Research. Digital Research adds many system utilities

T/MAKER III
Quick Reference Booklet
Personal Micro Computers, Inc.
Qume
MicroMate

and programming utilities to the package free of charge (including MAC, RMAC, and SID, which, if sold separately, would cost more than CP/M Plus), and provides a complete set of manuals covering their operation and use.

Bundled and Recommended Software

Bundled with MicroMate is CP/M Plus with the CP/M programming utilities, T/Maker III, Electric Webster spelling checker and CBASIC. Also included is a program (COPYFIL) for copying individual files from one floppy disk to another using a single disk drive.

MITE (Mycroft Labs, Tallahassee, FL) is recommended for telecommunications (*to be reviewed in a future issue — ed.*). Application programs designed for CP/M systems will run without modification on the MicroMate; however, the programs must be "installed" for your particular display screen.

T/Maker III

The MicroMate comes with T/Maker III, a program that can perform word processing, spreadsheet calculations, mathematical operations, list processing, data transfer, file management and bar charts. Predating Lotus 1-2-3, this program is one of the first integrated programs containing many different application tools that use the same data, and it is available for both eight- bit and sixteen-bit CP/M systems. We will be covering T/Maker III in a future article.

CP/M Plus Utilities

Utilities are programs you use to manage files and disk space.

The ERASE utility provides a quick confirmation check for each file before actually erasing it, giving you time for a second thought. You should put this utility and the RENAME utility on every system disk. This enhanced version of the REN command lets you rename several files at once using the *filename match symbols.*

The DIR utility is not necessary for simple directory displays, but with it you can display filenames in alphabetical order with the file sizes, attributes, protection mode and time/date information. The SHOW utility (replacing STAT) displays the amount of space left on a disk (DIR does not show this information, although many public domain programs, such as SD and XDIR, helpfully provide this information with the directory display).

COPYSYS is for copying the inner two tracks — the *system* tracks — of your system disk to another disk. This utility lets you turn a data disk into a system disk that can work in drive A if you include the CPM.SYS file on the disk.

DATE, INITDIR, and SET are useful at system startup to set the time and date, prepare a new disk for time and date information (INITDIR), and set various system options such as password protection, disk labels and file attributes. Default values, such as the pause during screen scrolling, are set for normal operations. To change these defaults, you need to execute SETDEF once for the system disk. You can set the system search order, and turn on-/off the display scroll pause, and change other defaults. The system disk retains the search order and screen information.

These utilities are sometimes combined into a "submit" file of commands to be executed all at once. You don't need them on every system disk, but you may need to use a "system warm-up" disk with these utilities preparing your system or preparing new disks.

MicroMate Utilities

In addition, PMC offers several utility programs for using disks and configuring devices. CONVERT lets you convert a disk drive to read/write different single and double density disk formats (as mentioned above). BACKUP is a simple disk copying utility for single-drive or multi-drive disk copying. COPYFIL is a utility for copying files from one disk to another using a single drive.

CONFIG is a utility to change the configuration of each port. You can change baud rates, start and stop bits, parity checking and protocol for both serial ports using this utility without making any hardware modifications or setting external or internal switches.

Using the System

Simple and useful utility programs are provided to handle files, prepare disks and make backup copies. CONVERT lets you convert a disk drive to read or write a different disk format. You are presented with a list of popular disk formats, including the MicroMate's own formats: double-density single-sided (200K), and double-sided double-density that can hold 400K.

FORMAT presents a simple menu to choose from: *F* to format the disk, *V* to verify that disk tracks are readable, and *E* to exit the FORMAT utility. The BACKUP utility is just as simple, and also verifies tracks as they are copied.

Communications

PMC recommendts, options and protocols while communicating. MITE offers six different communications protocols:

XMODEM

The "Modem7" protocol (invented by Ward Christensen and placed in the public domain) used on most remote call-up CP/M software exchange systems (RCPM systems). This is the most widely used binary-file transfer protocol for CP/M systems, and is the default protocol for MITE. All CP/M files can be transferred with error-checking for error-free transmissions.

CLINK

The original Mycroft Labs binary-file transfer protocol, to communicate with computers running the CLink or Crosstalk programs.

IBMPC

The Asynch. Support Package text-file protocol for the IBM PC. This protocol does not let you transfer binary files (files with ".COM" or ".CMD" file type extensions). No error-checking is performed, and only ASCII text files can be transmitted or received.

HAYES

The Hayes Terminal Program Verification Protocol.

MITE

The protocol used by MITE and by SEND/RECV, also from Mycroft Labs. This protocol lets you transfer multiple files at once (like the "batch transfer" command of Modem7) by specifying a filename match (such as "*.TXT").

TEXT

A simple ASCII text file transfer protocol that can communicate with most minicomputer or mainframe computer programs to transfer text files with error checking and recovery.

MITE's strongest feature is the ability to create a *macro* — a string of automatic commands and typed responses to automate the log-on procedure of most electronic information services. MITE can also dial phone numbers for you.

We expect the MicroMate to be useful as a transportable computer in an office environment where a terminal, usually connected to a mainframe or minicomputer system, can be connected quickly to the MicroMate, which in turn can be connected to the mainframe or minicomputer system. You start up MITE and you are instantly in communication with the host computer. You can "download" data on your disk to the host computer, and "upload" information from the host for storage in your disk.

Terminal and Modem Connections

With two serial ports and two parallel ports, the computer lets you connect a terminal, modem, extra disk drives and printer all at once and use them simultaneously (the ability to do this is essential for any serious computer user).

For a terminal we tried an ADDS Viewpoint, a Televideo 950, a Qume QVT-102, and a lap computer (the Radio Shack Model 100). All performed without a problem. We connected the MicroMate to another computer and were able to control the MicroMate from that computer (an Alspa ACI system running Modem7). It seems that any ASCII terminal, including the "pocket terminals" and hand-held computers, can be used with the MicroMate.

The CONFIG utility lets you configure your system as you need to, changing the terminal or modem baud rate or other transmission protocols. The MicroMate is supplied with several master disks to power up a system at different terminal speeds such as 110 baud, 300 baud, 9600 baud and 19,200 baud. This flexibility lets you move the MicroMate from terminal to terminal.

You set transmission protocols — the number of "stop bits" and whether parity checking is even, odd, or not used — in order to communicate with a variety of systems.

The DEVICE command can set the XON/XOFF protocol for any device. You can also set the system for RTS and/or DTS protocols by changing a word in a source file, and performing a SUBMIT with a supplied SUBMIT file (which automatically re-assembles the basic input-output, or BIOS, section of the system).

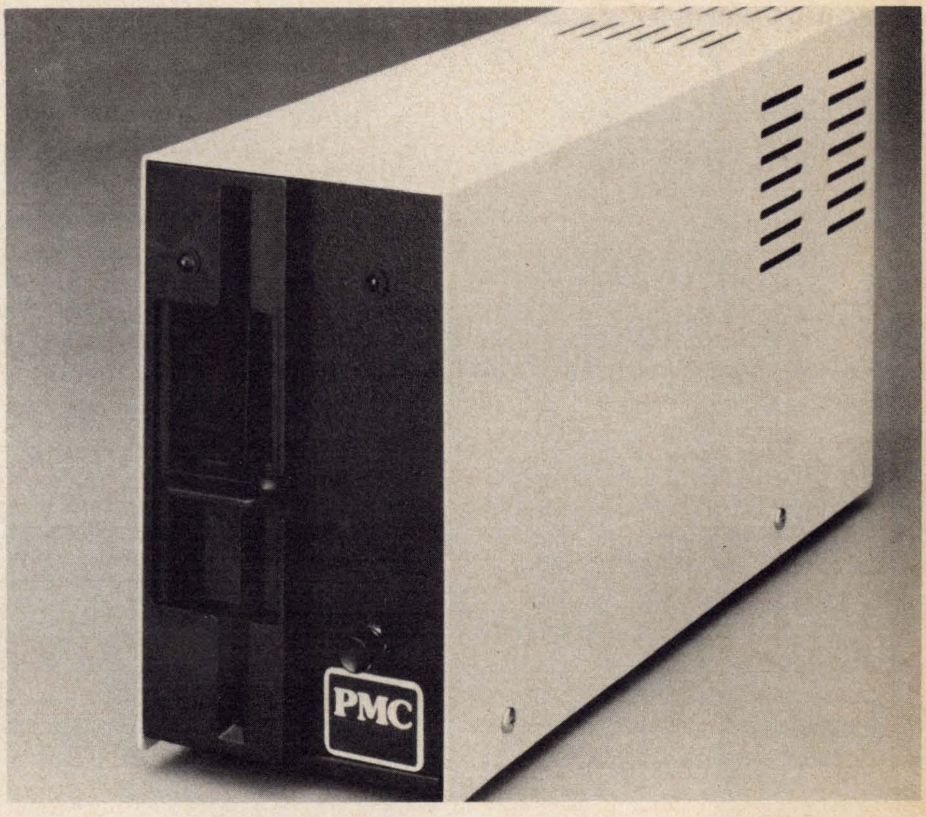
If you use different terminals, you can simply configure a different system disk to keep on hand for each terminal. Once the system is up and running, you can replace the system disk you used for start-up with another system disk or application disk containing the CPM.SYS file. Typing a Control-*C* (warm boot) to read the new disk's directory does not affect the terminal or modem characteristics, baud rates, or protocols.

Conclusion

The MicroMate works well, is flexible for connecting various terminals, and has more disk capacity in its single drive (using double-sided disks) than the two-drive Osborne or Kaypro computers that use single-sided drives.

For its price ($1195), it's the least expensive CP/M Plus machine, and it's less expensive than some used Osbornes or Kaypros we've seen for sale. You can pick from several excellent terminals for under $600, which is also the price of a Radio Shack Model 100. We really enjoy writing on the Model 100 and transferring files to the MicroMate. For traveling computer magazine writers, the combination has proved itself very useful.

The MicroMate works with any ASCII terminal, from the very inexpensive hand-held terminals to the most sophisticated CRTs.



Using the Radio Shack® Model 100 as a Terminal

The Model 100's TELCOM program can operate the computer as a terminal running at 19200 baud. We used TELCOM to connect the Model 100 to the PMC MicroMate. CP/M Plus lets you change device characteristics "on the fly" with the DEVICE command; e.g., you can change the terminal device to run at any baud rate, or to use the XON/XOFF protocol.

We did not need to use a terminal to set up the PMC MicroMate CP/M Plus system before connecting and using the Model 100 as a terminal for it. We used a system disk, set to respond to a 19,200 baud terminal with no protocol. The system started without a hitch when we pressed the reset button and displayed the sign-on message. The Model 100 screen did not actually scroll at that speed (19,200 baud), yet it also did not drop characters. The Model 100 screen wrapped all over-40 lines without losing a character.

Uploading From the Model 100

To upload text to the MicroMate, we used the CP/M Plus *DEVICE* utility with the *MODEM* option, to set the modem to operate at 1200 baud with XON/XOFF protocol:

```
A>DEVICE MODEM [1200] ⏎
```

In the absence of a modem program for the MicroMate (our version of MITE had not yet arrived), we used the PIP command with the *AUX:* device name to transfer "auxiliary input" (i.e., serial port "modem" input) to the file TEST.TXT:

```
A>PIP TEST.TXT=AUX: ⏎
```

We returned to the Model 100 TELCOM command level by pressing the function key to "quit" and disconnected by typing *Y*. Then we selected the upload option and sent the file out through the serial port. The PIP program accepted data until it received the Control-*Z* character signifying the end of the file.

See the review in this issue of the Model 100 for a description of the capabilities, and limitations, of using TELCOM for uploading.

Product Announcements...

TeleStar
©Copyright 1983 Portola Software

TeleStar® — for WordStar on the TeleVideo

Portola Software (Mtn. View, CA) announces TeleStar for TeleVideo and WordStar users. TeleStar is a software overlay for WordStar which includes 57 colored keycaps you replace on your TeleVideo keyboard. Documentation consists of an instruction guide that is color coded to the keyboard function groups.

The keycaps replace the 11 function keys, plus many others, making it easy for a novice to learn WordStar without having to memorize the control character sequences.

The software overlay includes an automatic SUBMIT routine which loads TeleStar onto your CP/M-80 (1.4, 2.2, or MP/M 1.1, 2.1) system disk which must also contain the WordStar 3.0 program and overlay files.

TeleStar retails for $150 or $200 for either 8 inch single density IBM format or 5 1/4 inch Televideo format disks.

Circle Reader Service # 75

Hard Disk for ACCESS Computer

Access Matrix Corp. (San Jose, CA), manufacturer of the ACCESS portable computer and Xebec (Sunnyvale, CA), disk drive controller and disk storage subsystem manufacturer, announce the Xebec 9800 10 megabyte 5 1/4 inch Winchester hard disk will be made available for $1395 as an optional external storage device for the ACCESS.

The ACCESS contains a dot matrix printer, acoustic coupler and direct connect modem, a 7 inch amber monitor, low profile detachable keyboard, two double density floppy disk drives (180K each) and computer with power supply all built into a single unit.

Circle Reader Service # 79

Hard Disks and Backup for IBM PC-XT

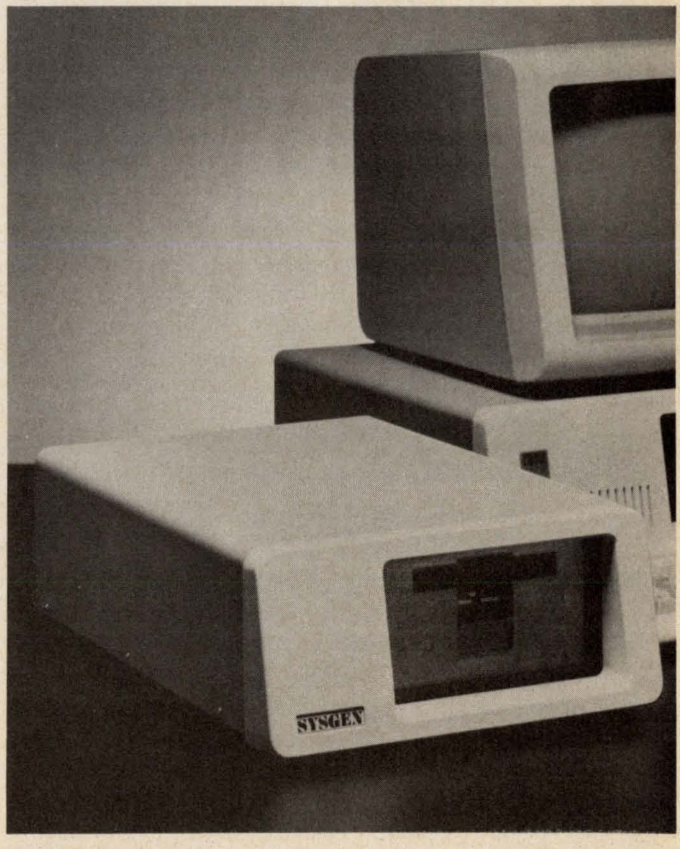
Sysgen, Inc. (Fremont, CA) announces the Sysgen Image®, a streaming cassette tape backup system for the IBM PC-XT. The suggested retail price of $995 for the subsystem includes controller, cables, software, drive electronics and tape drive.

The Sysgen Image is software compatible with either PC-DOS or CP/M-86 and holds 20 megabytes of prerecorded software on a tape cartridge included with the subsystem.

The Sysgen Image purchaser can browse through the "Softlok"ed demo programs that include a word processor, data base manager, spreadsheet program, and other programs written for the IBM PC. You "unlock" a program by contacting your dealer to exchange payment for the unlocking key of the program desired.

Other non-locked software gives you a choice of two backup modes to download with error checking from the PC-XT's hard disk. PRESERVE backs up a complete archival image at 2.5 megabytes per minute and a catalog utility scans the entire streaming tape to list all files by tapefile identification, size and other categories. FILESAVE saves individual or groups of files to the streaming tape or from tape to the hard disk.

Circle Reader Service # 76



Sysgen II IBM PC Winchester 10 or 20 megabyte hard disk system includes the Sysgen Image tape subsystem.

Continued on page 82

The Answer Man

Answers to Recent Correspondence and Advice

by Kelly Smith, Sysop
CP/M-Net (RCPM) 805-527-9321

I've had a chance to catch up on some answers to letters that appeared in UG volume 1 no. 4.

Learning dBASE II and BASIC

To: Paul Swift (San Mateo, CA), letter on p. 20 of volume 1, issue 4:

Contact your Osborne 1 dealer and ask if the six disk (OCC single-density) set of dBASE II Tutor is still available. This is a "limited implementation" of the full Ashton-Tate dBASE II package which includes a "system startup" and ten lesson set which allows a comprehensive self-teaching overview of dBASE.

As for learning BASIC (in your case Microsoft BASIC), I would suggest "downloading" any of innumerable examples in (source code form) various games, business, and scientific applications from your local RCPM. Try Byron McKay's Piconet RBBS-RCP/M at 415-965-4097. There are usually lots of goodies specifically for the OCC 1 for the price of a phone call. You'll need a modem, and as software for communicating with the RCPMs (Modem7 is described in UG volume 1 issue 3, "Downloading From RCPM Systems").

RCPMLINK, a friendly Modem7 alternative, is available from Wizard of OsZ (213-709-6969) for $50.00.

WordStar and Logged Disks

To: Janet Goldenberg (San Francisco, CA). Re: whether or not you replace a disk while still in WordStar.

An additional problem may arise on some computers that have not implemented a "buffer flush" in the CP/M BIOS disk "HOME" routine. This means that remnant data retained from a previous edit session with WordStar is in the computer's memory, which could be written to a new file (and excluded from the old file), if the original disk is not "closed" properly, which is what may happen in you swap disks while in WordStar.

A "warm boot" is recommended (Control-*C* at the CP/M command level) to insure that no "dirty buffers" are retained in memory.

Add Batch To SUBMIT

To: Don M. McLaughlin (Brandon, FL), Re: "batch processing" your DataStar files.

I assume your are using the supplied SUBMIT utility. Here is a short assembly language routine that can be included in your submit file (we call it WB.COM after assembly and loading), which will allow you to change data disks in drive B:, with a prompting "wait" message:

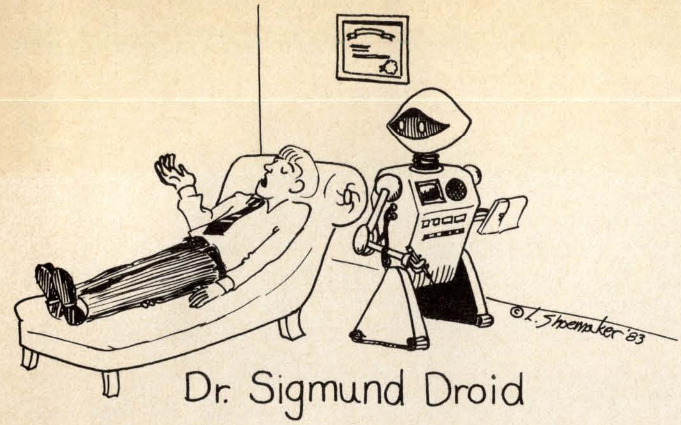
```
        org     100h      ; standard 'origin' address for .COM
                          ; files
        mvi     c,9       ; print message string
        lxi     d,msg     ; point to message
        call    5         ; let CP/M do the work
        mvi     c,1       ; console read (to wait)
        call    5         ; let CP/M do the work
        mvi     c,0       ; reset disk system
        call    5         ; let CP/M do the work
        ret
msg:    db 0dh,0ah,'Press any key after changing disk: $'
;
end
```

This routine will reset the disk system and allow you to change disks during your DataStar file processing (*don't change* the disk in drive A, since it will be processing your SUBMIT file!). I have not used DataStar, but you can probably rewrite the Scan Mask by using the XSUB companion utility to SUBMIT to generate the new mask for each disk change as if it was coming from the keyboard. Hope this solves your problem!

Latest Versions of Modem7 and MBOOT

To: Marvin E. Decker (San Antonio, TX) and W. Duane Austin (Mtn View, CA).

Little by little, customized versions of Keith Petersen's MBOOT (Modem Boot) program are appearing on many RCPMs to deal with the specifics of system hardware and serial data interface configurations — UARTs (Universal



Dr. Sigmund Droid

Asynchronous Receiver/Transmitters) and their associated hardware "port addresses."

There are versions of Modem7 and MBOOT for the popular UART devices. Most computer manufactures are using the Zilog Z80 SIO, Zilog Z80 DART, Intel 8251 USART, or National 2661 USART, so that given the specific port addresses, the MBOOT and Modem7 programs are transportable to most systems. For example, two of the latest customized MBOOT programs on my RCPM system are:

MBOOT-RB.ASM, DEC Rainbow 100 compatible MBOOT source OCXMBOOT.ASM, Osborne Executive compatible MBOOT source APMBOOT.OBJ, Apple][(with Z80 softcard) compatible MBOOT object

Look at the latest RCPM-xxx.LST (xxx is the issue number), which is the RCPM Systems "Phone Book" file, for the nearest RCPM system in your area. You'll find that the "special interests" of the SYSOPs cover most of the systems available with the specific modem software that you need for your computer.

My RCPM (CP/M-Net, 805-527-9321) is for the interest of OCC 1's and OCC Executives, as well as the DEC Rainbow 100. Trevor Marshal's RCPM (Thousand Oaks Technical RCPM, 805-492-5472) is for the Ferguson Big Board, Paul Kelly's RCPM (Superbrain RCPM, 617-862-0781) is for Superbrainers, Kim Levitt's RCPM (Hollywood RCPM, 213-653-6398) is for Kaypro users, and so on.

Let your fingers do the walking! P.S., As of this response to your letters, the latest version of Modem7 is actually Modem901. This program just keeps getting better all the time!!!

Flippy Disks

A letter of my own on making "flippies" (BDOS errors in the making!).

I am seeing numerous recommendations (and devious devices) for punching additional holes in floppy disk jackets to allow the user (abuser?) to "flip" the disk and gain access to the other side of the disk recording media. In a word, *don't*! The reasons why?

1. The "single-sided" disk has probably not been tested for the "flip" side.

2. Some manufacturers of disks do not perform a process called "burnishing," which is a polishing process to finish the disk surface. A cross-section of a non-burnished disk looks like the Andes mountains to a read/write head, and can be very abrasive (like sandpaper) on your drive's read/write head. I don't recommend head cleaning diskettes either!

3. The soft liner within the disk jacket is used to pick up microscopic particles of iron oxide that are normally shed from the media surface. They are needle-like in appearance and are driven into this liner. When you flip a disk, the rotational direction is then opposite than that which drove these "needles" into the liner. Where do they go? They are driven back out (like a herd of porcupines) onto the disk surface, causing abnormal were and tear.

4. The drive read/write heads have been optimized for the best "surface contact vs. wear" for what the disk manufacturer calls "penetration." This is identical to the "wrap" across the heads in an audio tape deck that requires very tight mechanical tolerances for the best recording/playback. Due to the folded edges of the disk jacket, by flipping the disk you are moving the diskette surface *away* from the read/write heads by as much as 20 thousandths of an inch (depending on the manufacturer, and the technique used to weld the jacket edges). This means that your drive cannot write as "hard" into the media, and "feeble reads" can occur in subsequent use.

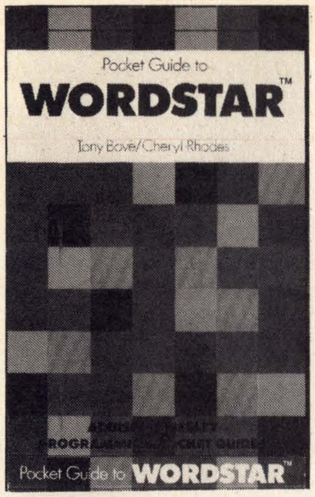
5. The original index and write protect holes (unprotect on 5.25-inch disks) are "hot punched" by the manufacturer. That is, the hole is self-sealing and welded at the edges. Punching additional holes cause microscopic fractures and splinters of plastic that float around inside the disk jacket along with the rest of the porcupines (see 3), to peel-off the delicate media coating.

One last comment, about protective hub rings. Due to poor registration of these "hole protectors," your disk will run eccentric (out-of-round) to the actual disk center intended. This becomes particularly sensitive on 96 track/inch disk drives (and double-density formats) where the track "center" needs to be right down the middle.

Pick up most any disk with a hub ring, and it will be misregistered by as much as ten-thousandths of an inch (i.e., about one track out of 96 tracks per inch). The mechanics of most disk drives these days are such that "disk crunching" has almost disappeared. So, don't pay more for a problem that in most cases does not exist and can only cause aggravation to your data.

Trust me, I have twelve years experience in disk drive engineering! *(Pretty convincing, huh? — Editors.)*

Impatient User's Guide to


Pocket Guide to
WORDSTAR™
Tony Bove/Cheryl Rhodes
Pocket Guide to WORDSTAR™

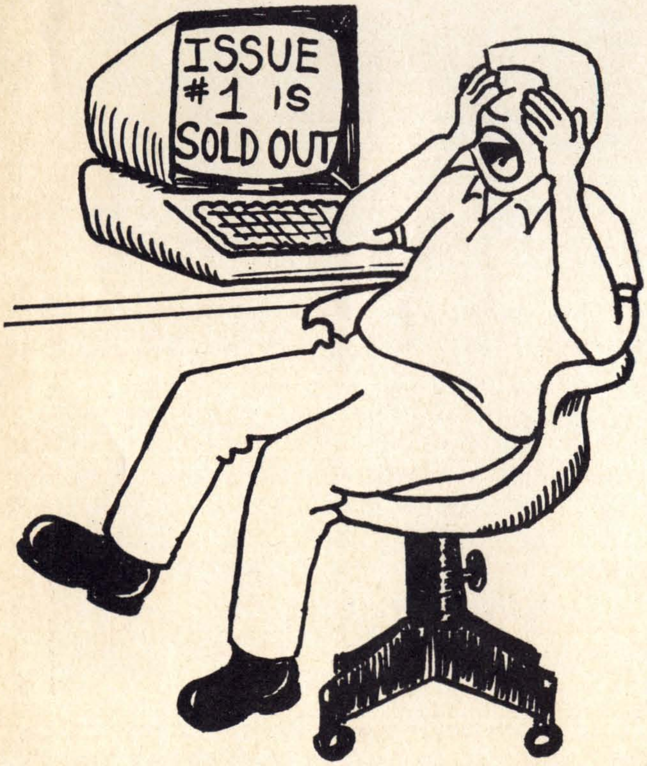
Actual size 4" x 6.5"

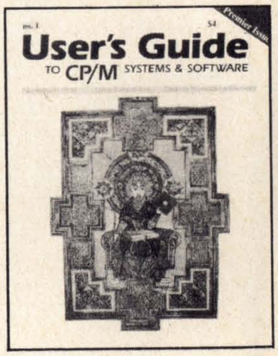
- CP/M-80 and CP/M-86
- WordStar
- SpellStar
- MailMerge

All-in-one pocket reference guide from Addison-Wesley, with an index so complete you won't waste time searching through manuals.

Ask for it in your local book or computer stores.

Issue No. 1 Sold Out!


ISSUE
#1 IS
SOLD OUT


no. 1
$4
Premier Issue
User's Guide
TO CP/M SYSTEMS & SOFTWARE

Thanks for being supportive — but unfortunately we are sold out of the first issue of *User's Guide*.

We are considering doing a full press run of a revised edition of the first issue, with additional material, and with a cover price of $5. If you are interested, please send requests (with no money) immediately to **We Want Number One Dept., User's Guide, PO Box 3050, Stanford CA 94305**.

CP/M 83 EAST

User's Guide will be at booth 309 at the CP/M 83 East show at the Hynes Auditorium in Boston. This show focuses on personal and small business computers and software using the CP/M family of operating systems. See you there!

A sampling of what you can expect ...

Issue #1
The World of CP/M
Learning to Use CP/M
Impatient User's Guide to CP/M
CP/M Quick Reference and Summary
Using CompuServe
Preventing Computer Disasters
The Osborne Portable Computer
The IBM Personal Computer
The Alspa ACI-1

Issue #2
For Starters: Word Processing and Spreadsheet Calculations
A Guide to Using WordStar® (Pt. 1)
WordStar Impatient User's Guide
WordStar Installation Guide
Disk Doctor: The Rx For Bad Disks
CP/M Tutorial: Modifying Your BIOS, Files and the Directory
A Guide to Using SuperCalc® (Pt. 1)

Issue #3
For Starters: Mailing Lists
A Guide to Using MailMerge®
MailMerge Impatient User's Guide
Public Domain CP/M Software
Indexing Documents with DocuMate®
Footnoting Documents With FOOTNOTE®

User's Guide focuses on business applications for CP/M systems. The following articles are planned for future issues (this list is subject to change):

Using SuperCalc (Part two): Personal Income Taxes
Impatient User's Guide to SuperCalc
Using WordStar (Part two): Advanced Editing and Printing
Comparing the Portable Computers: Support and Performance
Public Domain Software: MODEM7, WASH
CP/M "Shell" Programs: CP+, Power, others
For Starters: Understanding Languages
Overview of CP/M Languages
Comparing Pascal Language Implementations
Introduction to PL/I and C
IBM Personal Computer with Concurrent CP/M-86
Overview of CP/M Hard Disk Subsystems
Public Domain Software: Bulletin Board Systems
CBASIC, CB-80, CB-86: Applications
Programmer's Helpers: Display and Access Managers
For Starters: Data Base Management Systems
Designing a Database With dBASE II
Comparisons: FMS-80, dBASE II, others
DIALOG: A Researcher's Database
Introduction to The Source
Accounting Plus: Business Management
Financial Planning: FPL, T/MAKER III, others
Introduction to Multi-User Systems and Local Networks
The Bottom Line Strategist

You will have to buy your own User's Guide.

User's Guide
TO CP/M
SYSTEMS & SOFTWARE
MailMerge
Writing Tools
Typesetting
MODEM7
Petit Parisien

If you are one of the million+ users of CP/M systems and software, and you don't have a complete set of *User's Guides*, order back issues today!

Issue no. 1 is sold out! Only 12,000 people received a copy of this valuable collectors item. Issues 2 and 3 are going fast — don't wait! Use the subscription card in this issue to order back issues at $4 each, while they last!

If you are interested in article reprints, write to: Reprints Dept., User's Guide, PO Box 3050, Stanford, CA 94305.

Continued from page 77

Tallgrass Technologies (Overland Park, KS) is manufacturing two new Winchester hard disk drives for the IBM PC which include one of their two new streaming tape cartridge backup subsystems.

The Winchesters store 35 megabytes of formatted data for $5595 or 70 megabytes for $6595, and include a 45 megabyte user selectable streaming or file by file tape cartridge backup subsystem.

Both new tape cartridge subsystems emulate IBM's BACKUP and RESTORE utilities. The 20 megabyte system retails for $2195 and the 45 megabyte system retails for $2495.

Circle Reader Service # 77

Mountain Computer Inc. (Scotts Valley, CA) announces a 10 megabyte cassette tape backup subsystem called the Image® compatible with IBM PC and Apple computers with 5 1/4 inch Winchester hard disk drives. The controller costs $295 and the tape drive costs $1095. The single-head, four track cassette drive can record or transfer at either 30 or 90 inches per second, with a backup time of 12 or 4 minutes.

Circle Reader Service # 78

Programming Languages

Digital Research (Pacific Grove, CA) has announced plans to support PC-DOS by offering languages that include CBASIC Compiler with graphics, Pascal/MT+, PL/I, C and Level II Cobol, and productivity tools that include Access Manager, Display Manager, Programmer's Utilities and Symbolic Debugger.

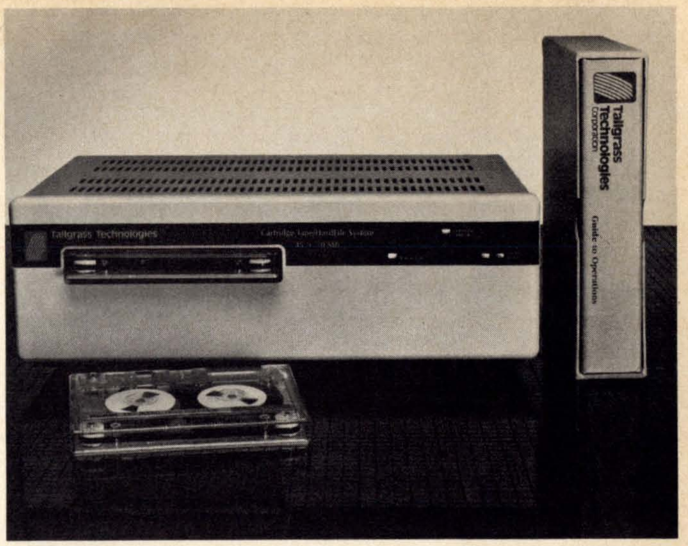
By supporting such popular operating systems as CP/M, Concurrent CP/M-86 and PC-DOS with their languages and productivity tools, Digital Research is strengthening its commitment to software developers (ISV's) by providing them with products that allow them to reach the largest possible market in the shortest amount of time with their application software.

All products are available now. Level II Cobol (Micro Focus) is $1600, PL/I is $750, CBASIC Compiler with graphics, Pascal/MT+ and C are each priced at $600. Access Manager is $400, Display Manager is $500, Programmers's Utilities package is $200, and the Symbolic Debugger is $150.

Circle Reader Service # 80

Utility Programs

Heritage Software (Los Angeles, CA) has released a new version of Smartkey, a background utility program you use to customize your keyboard.



Hard disk with tape backup system from Tallgrass.

Smartkey II is compatible with CP/M-86, CP/M Plus, and MS-DOS "and all known applications software including WordStar and dBASE II" says Heritage Software.

User's Guide has received the latest version of Smartkey II and will review this utility next issue.

Circle Reader Service # 81

Utility for Hard Disk Backup

Computer Dynamics (Greer, SC) announces "The BAKUP Package", a set of utilities used for file archival operations for Z-80 based computer systems with CP/M 2.2 hard and floppy disk storage.

For $50 you get an 8 inch floppy disk containing many utilities for making backup copies of hard disks onto floppy disks, automatic fragmenting of files too large for one floppy disk, a utility for restoring archived floppy disk files back to the hard disk, a global compare utility, an extended disk directory utility and more.

Contact Computer Dynamics for information about "The BAKUP Package" and Winchester hard disk systems with CP/M-80.

Circle Reader Service # 82

Ficomp, Inc. (Fairfax, VA) announces the BAKUP package of machine language programs for Z-80 based CP/M 2.2 computers. The set of programs are used to back-up files on systems using both hard disks and floppy disks.

BAKUP is available for $49.95 on 8 inch single sided IBM format or 5 1/4 inch Xerox 820, KayComp II, Televideo 802, Cromemco 520, DEC VT-18x, and Telcon Zorba computer formats. A special version is available for the limited screen Osborne 01. "These utilities are not public domain", says Ficomp.

Circle Reader Service # 83

WIZARD of OsZ™

a division of ™ Contact Enterprises

New Releases in Stock and Immediately Available ™

'Software and Accessories for Osborne™ Computers is our *only* business'

CHEKWIZ™ — the ideal financial tool for home or office
Handles over 1400 check entries and allows quick search by category, date, or name. Search and itemize a year's checks; ideal at tax time. Handles amounts up to 99,999,999.99. Allows easy entry of service charges, interest, automatic withdrawals, and provides automatic backup to protect against inadvertant loss of data. . . . . List $50

PLINK™ & RCPMLINK™ — now on one disk
Transmit complete file to bulletin board in seconds with user friendly PLINK. Saves files to buffer and writes to drive A or B. RCPMLINK enables fast, accurate block transfer of files between remote CP/M systems (or other Osborne 1 Computers). List $50

SIGNALMAN™ MARK V MODEM — Source℠ subscription included
A small, light, compact direct connect modem for Osborne 1™. Just plug it in and communicate through any modular telephone. Complete with cables and 9-pin Osborne wired connector. Uses 9v internal battery; 9v dc adapter is available. . . . . List $129

GAMES DISK 4 — graphics at play
2DMAZE, 3DMAZE, BOUNCE and LANDER (by Mirashanta). Programmed in CBASIC. Games are in two levels, simple then advanced. All include extensive graphics. . . . . List $50

GAMES DISK 5 — from Casino to Deep Space
Play Black Jack against the computer, the house limit is $10,000. Or how about a return to the old west to try your hand as a miner? Goldmine has all the rewards and risks you can handle. Then jump into the future with an exciting game of Space War. For a real challenge, try to win at Wizdot. Match wits with the computer, add line segments to make squares. Make 5 and you win. . . . . List $50

GAMES DISK 6 — educational fun from pre-school to college
Ghost in the Chimney delights pre-schoolers with visible rewards for right answers. Dingo the Cookie Goblin teaches addition and subtraction in a subtle form. Supermind pits your skills against the computer which hides 5 random letters; you have 13 tries to guess the letters in order. . . . List $50

Write or Call WIZARD OF OsZ™ Your Source for Osborne™ Software

VISA Add $2.50 shipping & handling Dealer Inquiries Invited Calif. add 6½ % sales tax master charge

9614C Cozycroft Ave. Chatsworth, California 91311 Telephone: (213) 709-6969

Circle number 19 on the reader service card.

Videotex Network News

Compuserve has added a new electronic Bulletin Board service targeted to the interests of the members of the American Society for Computers in Medicine and Dentistry (ASCMD), a group of health care professionals with an interest in the application of computers to both their professional and personal lives.

The principal goal of the society is to further the exchange of ideas and computer usage in the nation's medical community. The society ALSO publishes a quarterly newsletter for its members.

The ASCMD service is open to all interested Compuserve subscribers for the standard fee of $5 per hour from 6 p.m. to 5 a.m. weekdays and all days weekends and holidays.

Circle Reader Service # 84

MEDSIG added to Compuserve

The American Association for Medical Systems and Informatics (AAMSI) is a nonprofit professional association located at 4405 East-West Highway, Ste. 402, Bethesda, MD 20815.

Its members are concerned with developing scientific and educational programs and disseminating information; promoting the development of medical systems and information on patient care, teaching, research, health administration and fostering medical computing and information technology.

Contact AAMSI for details on accessing their Medical Forum (used to transmit messages among health care providers, information specialists and researchers) and their Communications Section (used to exchange information including clinical protocols, software, systems specifications and configurations and other biomedical computer-related analyses) available through the Compuserve Information Service.

Circle Reader Service # 85

CP/M Color Computer System

Memotech Corp. (Denver, CO) announces a new CP/M based color computer for $595. The MTX-512 premiered at a recent Computer Fair in London, England.

Many of the business programs sold by Memotech are presented in full color (WordStar, Graph Talk, etc.), as are the Oxford BASIC graphics (developed at the firm's research facility in Oxford) and many games.

Storage options include up to 4 5 1/4 inch floppy double sided double density drives (500K each), and Winchester hard disks with minimum 5 megabyte and maximum 80 megabyte storage.

80K of user RAM includes 16K dedicated video RAM. Maximum RAM is 512K. Graphics resolution is 256x192 pixels with 16 colors available high resolution. Although a 40 column display (color TV and video monitor output) is standard, an 80 column color RGB expansion option is available.

Standard outputs include: Centronics type printer port, two game controller ports (joysticks), separate TV and video monitor ports, a dedicated games cartridge port, a 2400 baud cassette port, an RS-232 port, and three voice sound plus white noise generator.

Circle Reader Service # 86

Advertise in issue 6 of **User's Guide**

National Advertising Sales Representatives:

Northeast (617) 444-3946 Silvio Mandino
Hajar Associates
280 Hillside Ave.
Needham Heights, MA 02194

Mid Atlantic (201) 741-7744 Pamela S. Griffith
Hajar Associates
321 Broad St.
Red Bank, NJ 07701

Southeast (305) 628-3525 Beverly Ellis
Hajar Associates
Diplomat Bldg.
5400 Diplomat Circle, Suite 272
Orlando, FL 32810

Midwest (312) 966-0160 Louanne Kohr
Hajar Associates
5225 Old Orchard Rd., Suite 50
Skokie, IL 60076

Northwest (415) 964-0706 Judith Blake
Hajar Associates
1000 Elwell Ct., Suite 124
Palo Alto, CA 94303

Southwest (714) 540-3554 Barbara Jones
Hajar Associates
3303 Harbor Blvd., Suite H-4A
Costa Mesa, CA 92626

• Reach an exclusive audience.

User's Guide readers own/use CP/M or IBM PCDOS systems in their businesses and professions. To get the most from their systems, readers are adding on software, peripherals and accessories. These business and professional users make buying decisions on new microcomputer systems, software, peripherals and accessories for themselves and their associates.

• Excellent residual pulling power.

User's Guide is sold in computer stores internationally, even in stores that do not carry other magazines. Why? It has useful content not found in other magazines. Stores carry back issues of *User's Guide* because the content covers major application software in-depth. *User's Guide* puts "hands-on" information at the fingertips of beginners and intermediate users.

• Reserve space now for issue no. 6.

The theme of issue no. 6 (Oct.-Nov.) is *data communications* The space deadline is Sept. 8. Camera-ready materials deadline is Sept. 15. Call your local Hajar Associates representative today for a media kit with current advertising rates, or contact us: *User's Guide*, PO Box 3050, Stanford, CA 94305 (415-851-4836).

• Low rates and a highly-qualified audience.

The low rates, combined with the highly-qualified audience of CP/M and PCDOS system owners (and users of WordStar, dBASE II, SuperCalc and other popular packages), make *User's Guide* your best advertising buy.

User's Guide **Volume 1, Number 5**
Sept. 1983.

User's Guide , ISSN 0745-4791, is published six times a year by TUG, Inc., Box 3050, Stanford, CA 94305. Second-class postage paid at Redwood City, CA 94063 (USPS 689-550).

Entire contents Copyright © 1983 by TUG, Inc. All rights reserved. Reproduction of material appearing in *User's Guide* is forbidden without explicit permission. Send all requests to The Editors, Box 3050, Stanford, CA 94305.

SUBSCRIPTION INFORMATION:

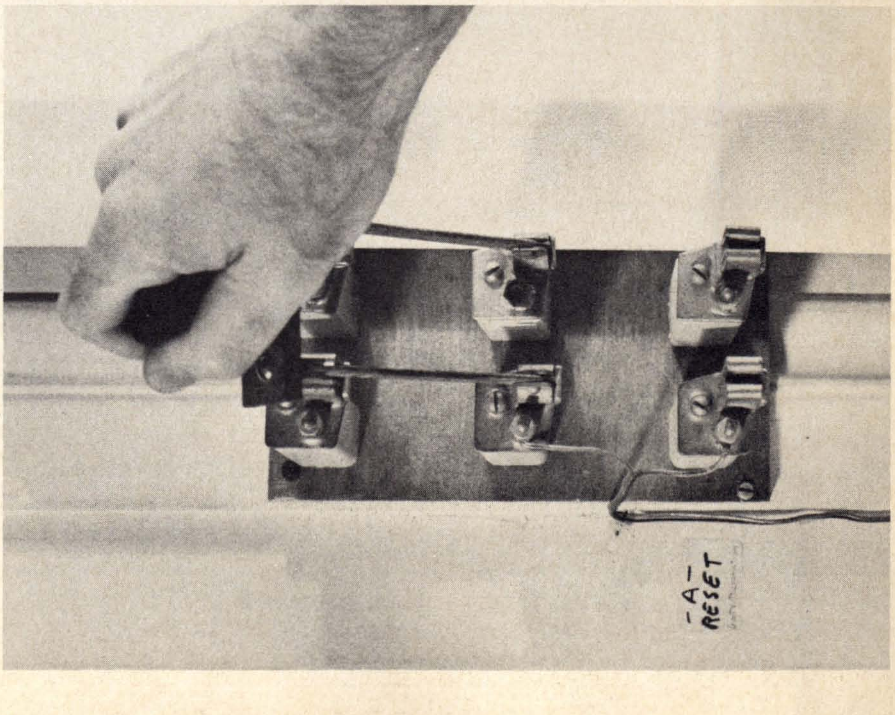
Single copy price $4.00 ($5.50 sent first-class). Subscription for one year is: U.S. — $18 if payment accompanies order, $21 if subscription is billed. Add $10 for first-class delivery.

FOREIGN - $24 U.S. for one year (surface mail) if payment accompanies order, $29 U.S. if subscription is billed. Inquire about rates for airmail delivery outside the U.S.

Inquiries about bulk, retail sale, and group rates are encouraged. Address subscription inquiries to *User's Box 3050, Stanford, CA 94305. Allow 6-8 weeks for processing of order.*

POSTMASTER: Please send address changes (Form 3579) to User's Guide, Box 3050, Stanford CA 94305.

(c) 1983 Rose Skytta



No... "Wait! A last-minute reprieve from the Master Control Program!" is not a good caption, is it? Let's have a **Caption Contest:** *send your entries to Caption Contest, User's Guide, PO Box 3050, Stanford, CA 94305.*

IS YOUR NEW CP/M™ MICRO STILL INTIMIDATING YOU?

By now, you may have discovered that CP/M is a wonderful operating system. But not an easy one to work with.

Here at last is **POWER!**, a super-power-packed, user-friendly program that takes the frustration and intimidation out of using CP/M.

POWER! offers 55 prompted, easy-to-use CP/M utilities in one 15k package. It automatically numbers disk files. From a screen menu, you pick the number to Copy, Erase, Reclaim, Rename, Type, etc. Your computer feeds the file names automatically. No more typing errors. And you can even run programs from the numbered menu.

No more BDOS errors. And you don't need a system disk in any disk drive!

POWER! lets you Test and Fix bad disks. Reclaim accidentally erased files or programs. Single step through memory up or down. Search, View, Change memory or disk in a snap. See Status and File Size instantly. Verify Checksums for programs. Load or Save programs at any address. Convert Hex numbers to Decimal or Binary instantly.

Now, **POWER!** also includes a special program that lets you lock sensitive files, so that only you can access them. Without the secret PASSWORD which you can create and change at will, no prying eyes will ever know your secret file even exists.

POWER! will operate in any standard CP/M or MP/M system, including CP/M-86, IBM PC, Apple (Z80 card), Osborne, Kaypro, TRS-80 conversions, S100's including NorthStar, Vector, Morrow, CompuPro, HP, TeleVideo, etc. Available in all disk formats.

It comes with a 120-page easy-read guide. And at only $3.00 per utility command, **POWER!** is your best buy in software. It's 55 friendly programs rolled into one to simply put you in control of CP/M.

Don't be intimidated by CP/M. Call or send in your order today and **Take The POWER! Trip.**

Only $169. Money Back Guarantee

Charge & COD Orders Welcome VISA MasterCard

TOLL FREE (800)428-7825 Ext. 96F
IN CA: (800)428-7824 Ext. 96F
DEALERS AND OEM'S (415)567-1634 Ext.F

InfoWorld
Software Report Card

Power Version 2.55

	Poor	*Fair*	*Good*	*Excellent*
Performance	☐	☐	☐	☒
Documentation	☐	☐	☒	☐
Ease of Use	☐	☐	☐	☒
Error Handling	☐	☐	☐	☒

© 1982 by Popular Computing, Inc. A subsidiary of CW Communications Inc., Framingham MA. Reprinted from InfoWorld.

Take The 'POWER!' Trip.

COMPUTING!
2519 Greenwich San Francisco, CA 94123

TOLL FREE (800)428-7825 Ext. 96F
IN CA: (800)428-7824 Ext. 96F
DEALERS AND OEM'S (415)567-1634 Ext.F

ONLY $169. Calif. add 6½% sales tax

☐ **CP/M $169.** ☐ **CP/M-86 $169.** ☐ **MP/M $249.**

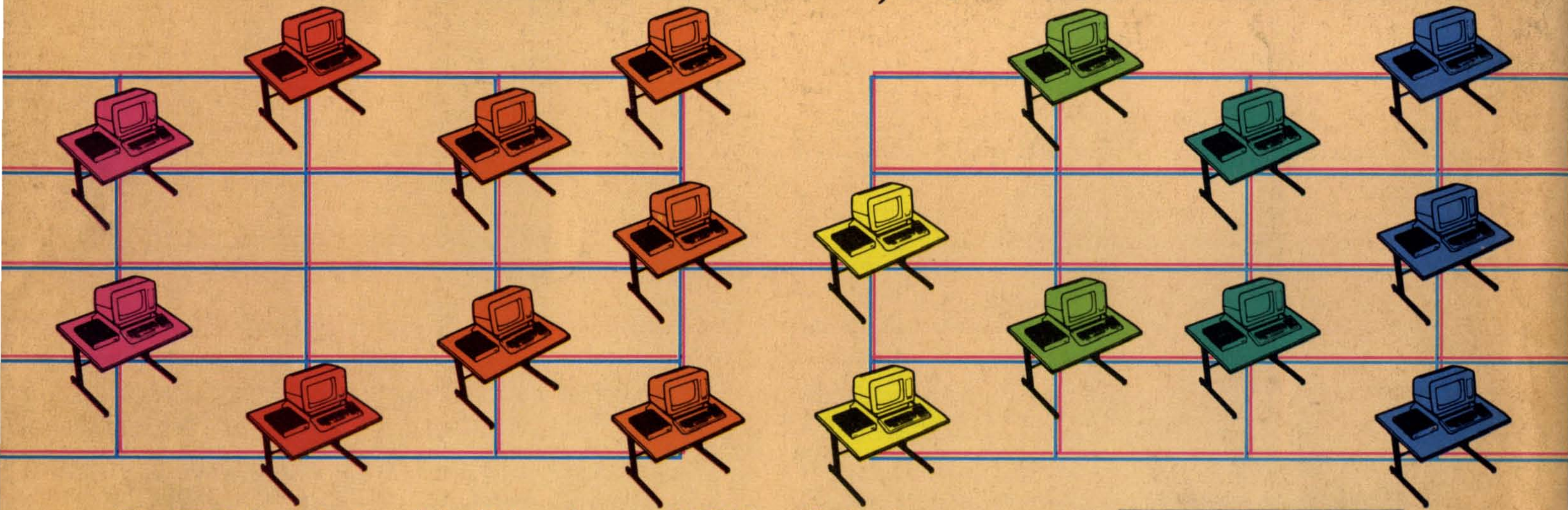
Card No. __________
Exp. Date __________
Computer __________
Your Name __________
Company Name __________
Address __________
City/State/Zip __________

Circle number 4 on the reader service card.

Alspa

ZERO

ANNOUNCING THE HIGH SPEED, LOW COST NETWORK



The **ZERO** is designed to bring high performance LOCAL AREA NETWORKING to users at budget prices.

The **ZERO** and **ZERO-NET** are unique. Any **ZERO** station can be a Network Master or Network Remote, permitting, for the first time, a low cost **non stop** network.

The **ZERO-NET** features a High Level Data Link Controller (message synchronous) at 400K bps carried over a simple twisted pair cable. To achieve maximum speed and reliability we use collision detection/avoidance circuitry and automatic CRC error detection/retransmission.

Each **ZERO** computer in the net can have Floppy Disk and/or Winchester Drives. The **ZERO** itself is a Z80 based Microcomputer with 64K Ram, 2K to 16K of EPROM, 2 Serial ports, 2 parallel ports, floppy disk controller, Z80CTC counter-timer and Z80DMA direct memory access.

The **ZERO** hardware design was optimized for TURBODOS*, (CP/M**, MP/M** compatible) including such enhancements as console type-ahead (buffering), 1.416 Mbytes per 8 inch double-sided floppy, multi-processing (background processing) such as print spooling, etc.

KEY PARAMETERS

- Local Area Networks up to 256 nodes per NET, with any mix of Master and Remote stations. Each station may support up to 16 logical drives, local or remote.
- Local Area Networks may be linked through gateways.
- Per Node — 0 to 2 floppies and 0 to 4 hard disks with appropriate Driver Modules.
- Per Node — parallel and/or serial printer.
- Each user can control print routing and/or spooling.
- Each node may reference a file system and/or printer on any other node.
- Each node may have an Autostart Log-on with security access protection.
- Each node may have a FIFO type Electronic Mailbox.

- Each node may operate with MP/M compatible file/record interlocks, or with special TURBODOS relaxation rules.
- Maximum recommended buss length of 4,000 L.Ft.
- Full CP/M and MP/M compatibility.
- The TURBODOS operating system can support up to one GIGA Byte (1,000 Mega Bytes) per logical drive.

The **ZERO-NET** product family — the **ZERO**, the **ZERO/FD**, the **Z-DRIVE** and the **ZNT** terminal — all the components required to configure an entire system are available for immediate delivery. The **ZERO** provides a microcomputer network that OUTPERFORMS many multi-user minicomputer systems at substantially lower cost. While the competition is still studying it, **ALSPA** has done it!

ZERO-NET

Alspa Computer, Inc.

300 Harvey West Boulevard, Santa Cruz, CA 95060
(408) 429-6000 Telex 176279

Circle number 1 on the reader service card.

---

*TURBODOS is a trademark of Software 2000, Inc.
**CP/M and MP/M are trademarks of Digital Research, Inc.